

---

# Breadth-First Search / Traversal

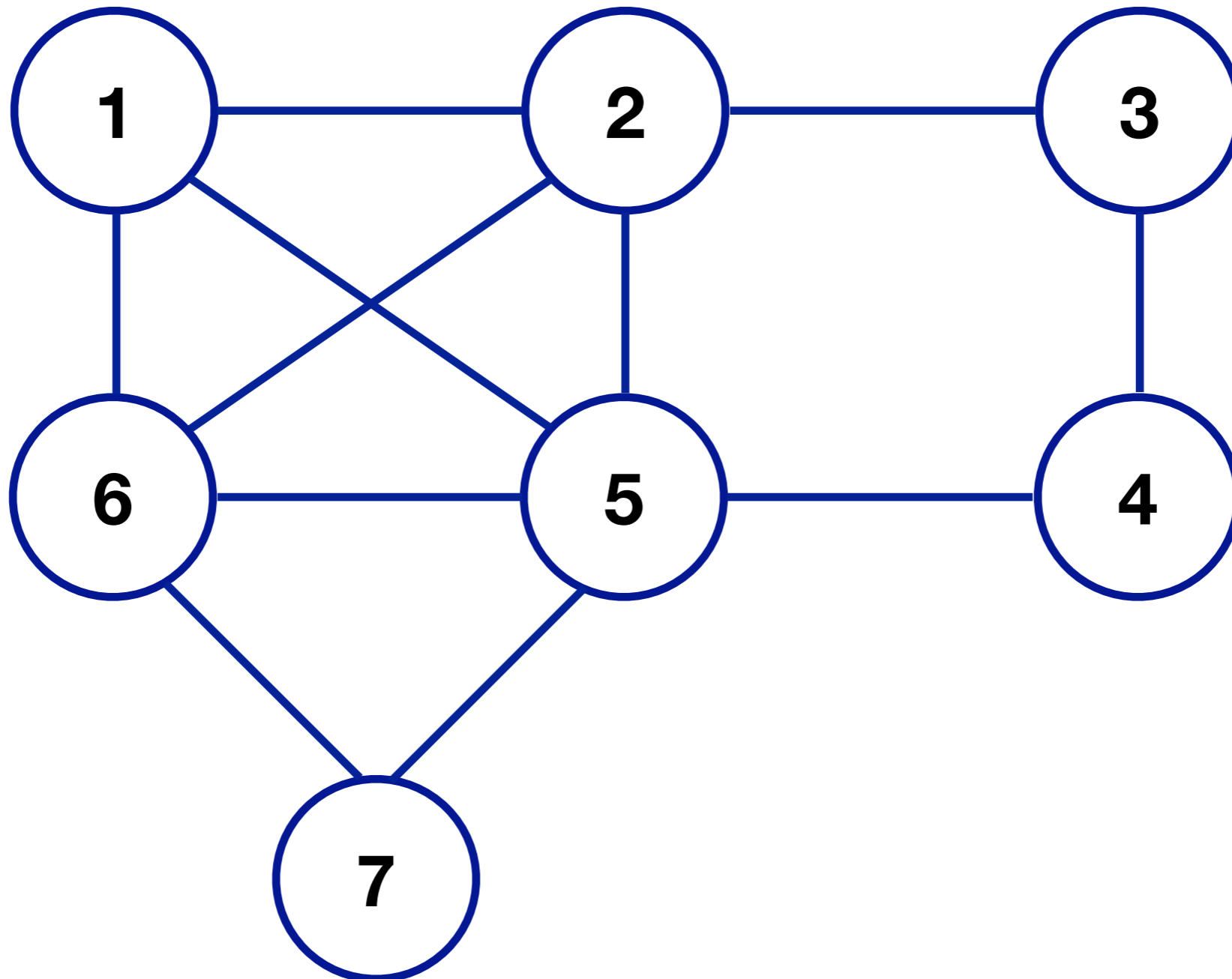
---



Alan G. Labouseur, Ph.D.  
Alan.Labouseur@Marist.edu

# Graphs

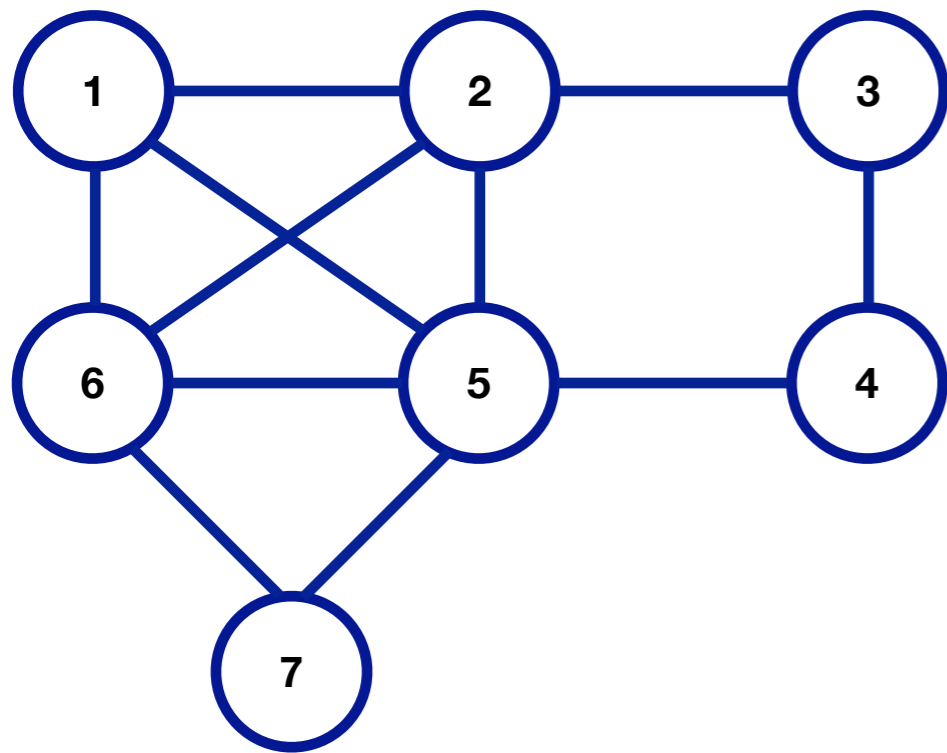
---



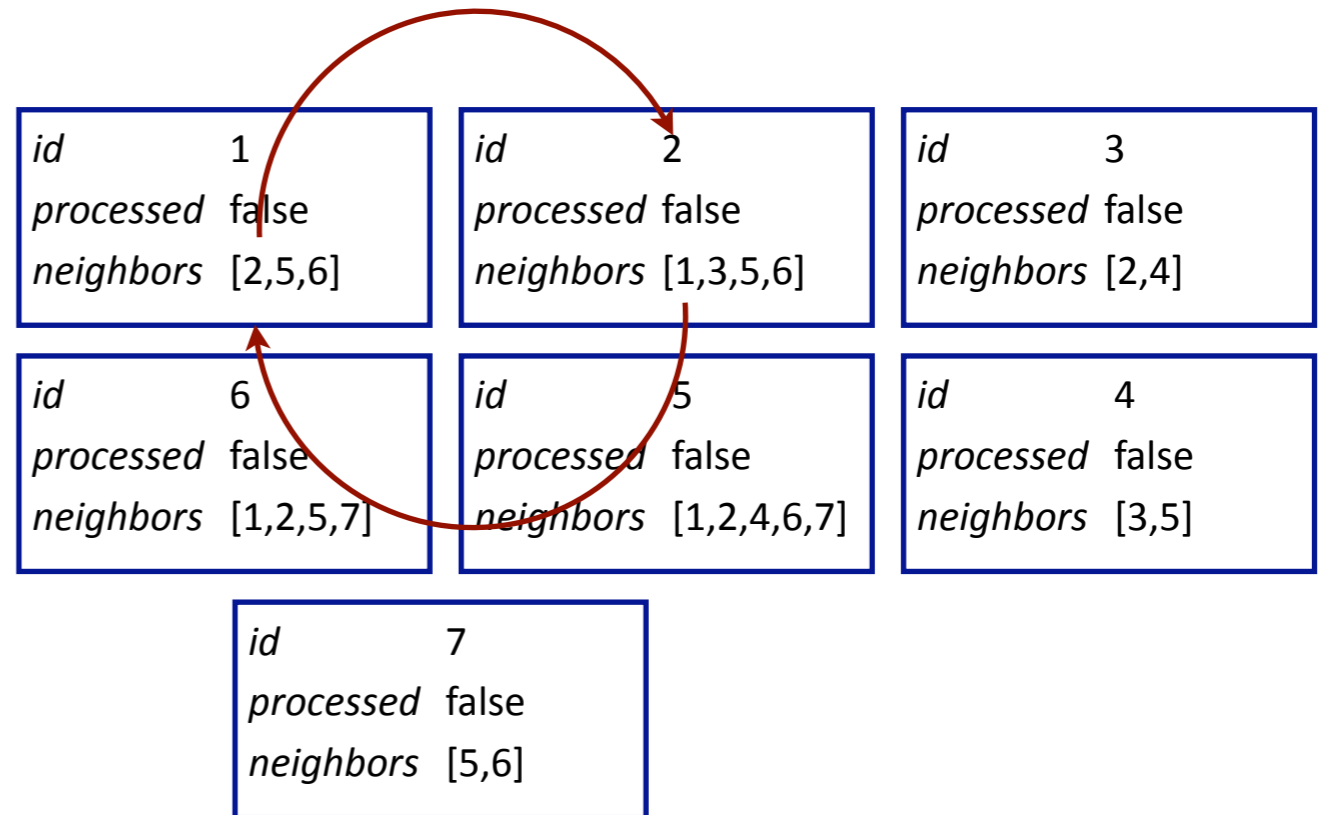
# Graphs

---

Graph . . .

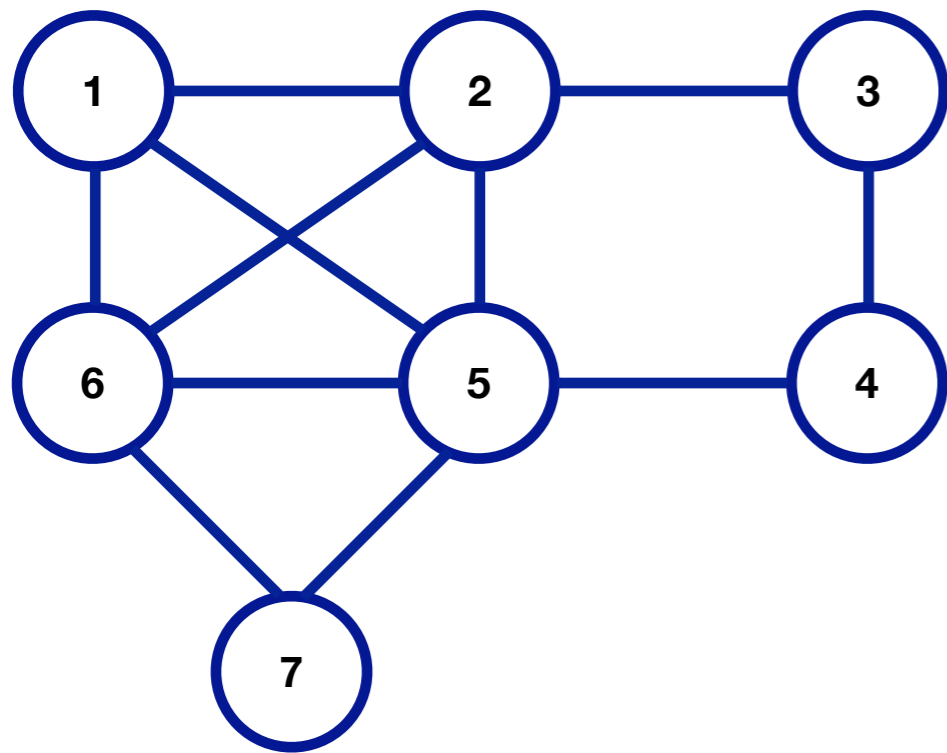


as Linked Objects

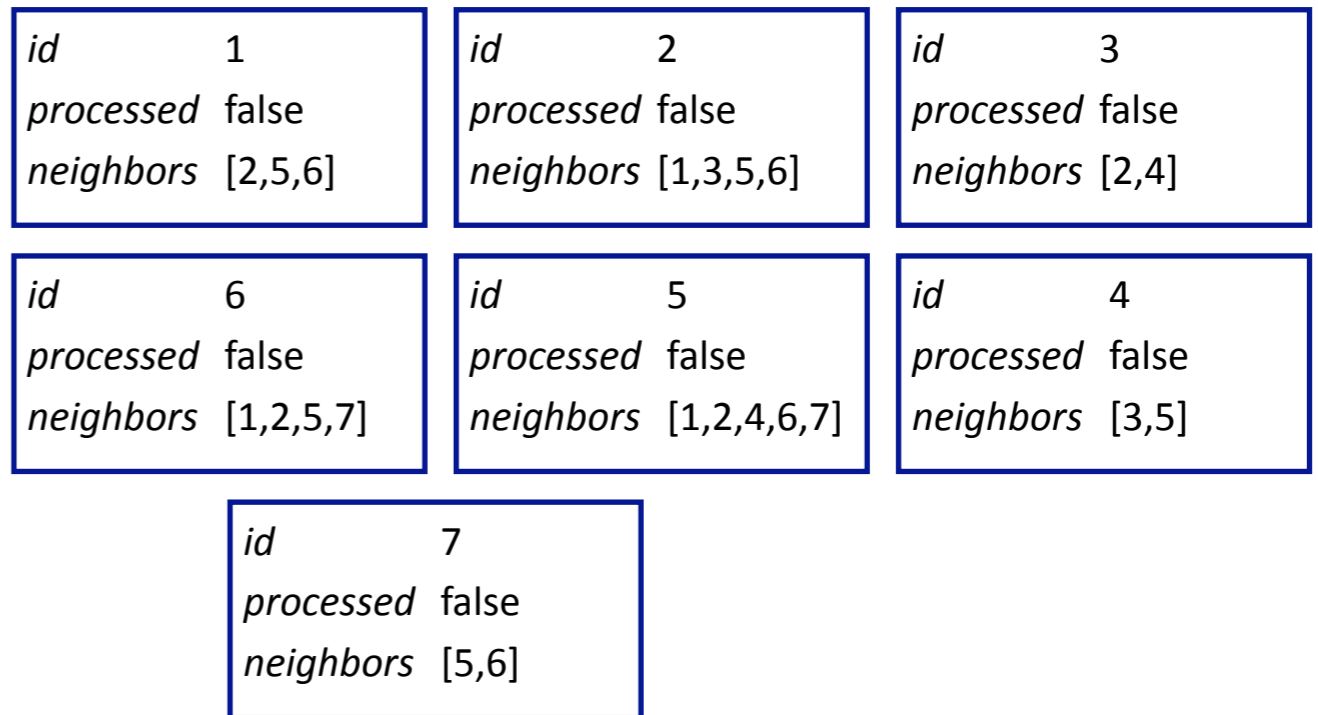


# Graphs

Graph . . .



as Linked Objects

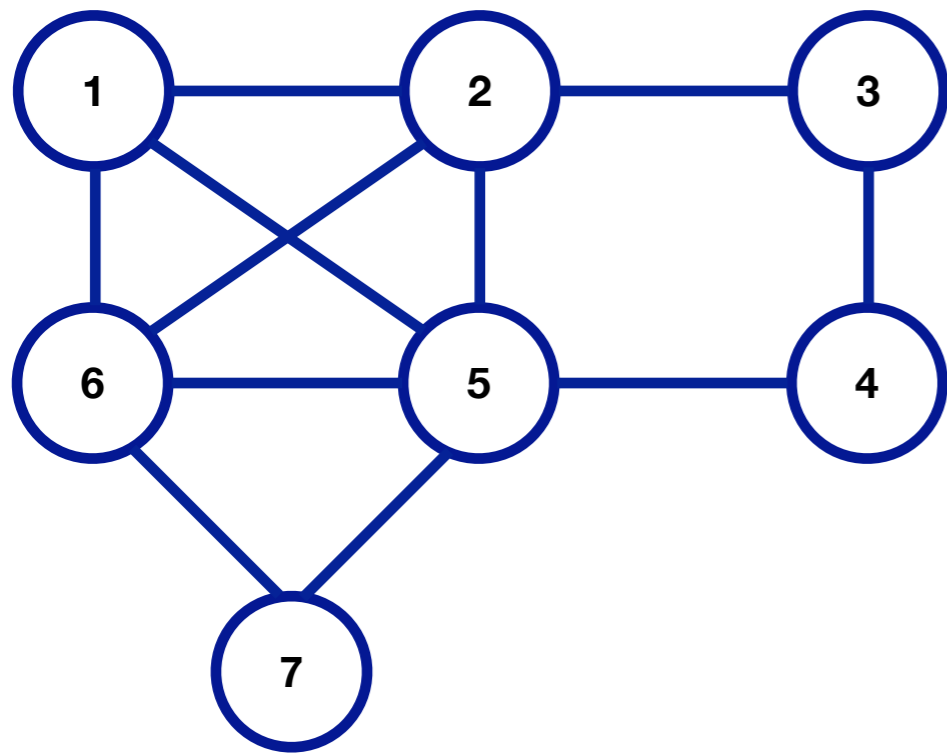


Adjacency List

```
[1] 2 5 6  
[2] 1 3 5 6  
[3] 2 4  
[4] 3 5  
[5] 1 2 4 6 7  
[6] 1 2 5 7  
[7] 5 6
```

# Graphs

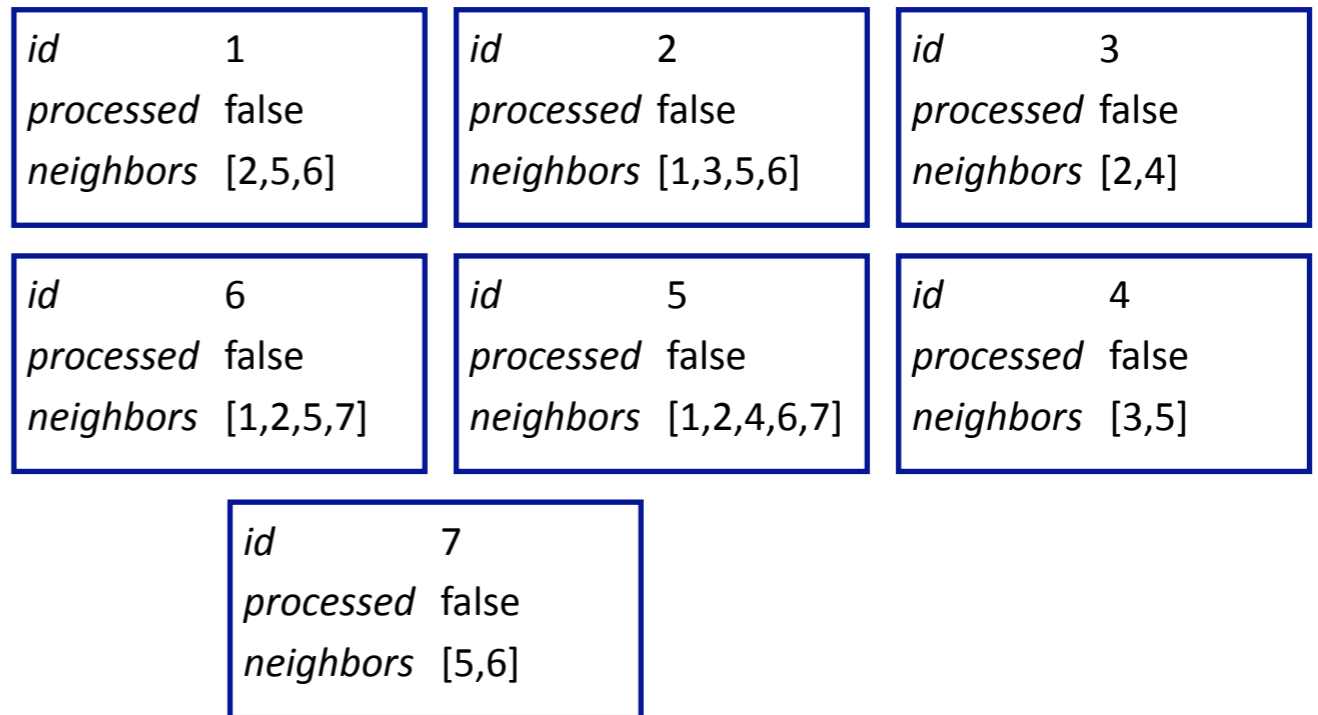
Graph . . .



Adjacency  
List

```
[1] 2 5 6  
[2] 1 3 5 6  
[3] 2 4  
[4] 3 5  
[5] 1 2 4 6 7  
[6] 1 2 5 7  
[7] 5 6
```

as Linked Objects

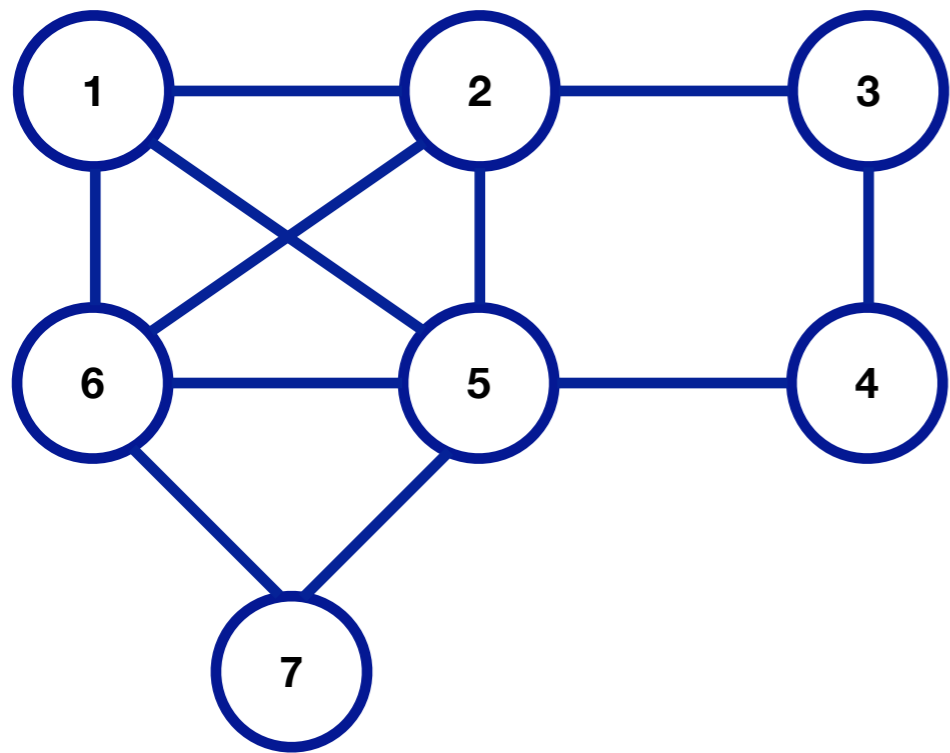


Matrix

```
  1 2 3 4 5 6 7  
1 . 1 . . 1 1 .  
2 1 . 1 . 1 1 .  
3 . 1 . 1 . . .  
4 . . 1 . 1 . .  
5 1 1 . 1 . 1 1  
6 1 1 . . 1 . 1  
7 . . . . 1 1 .
```

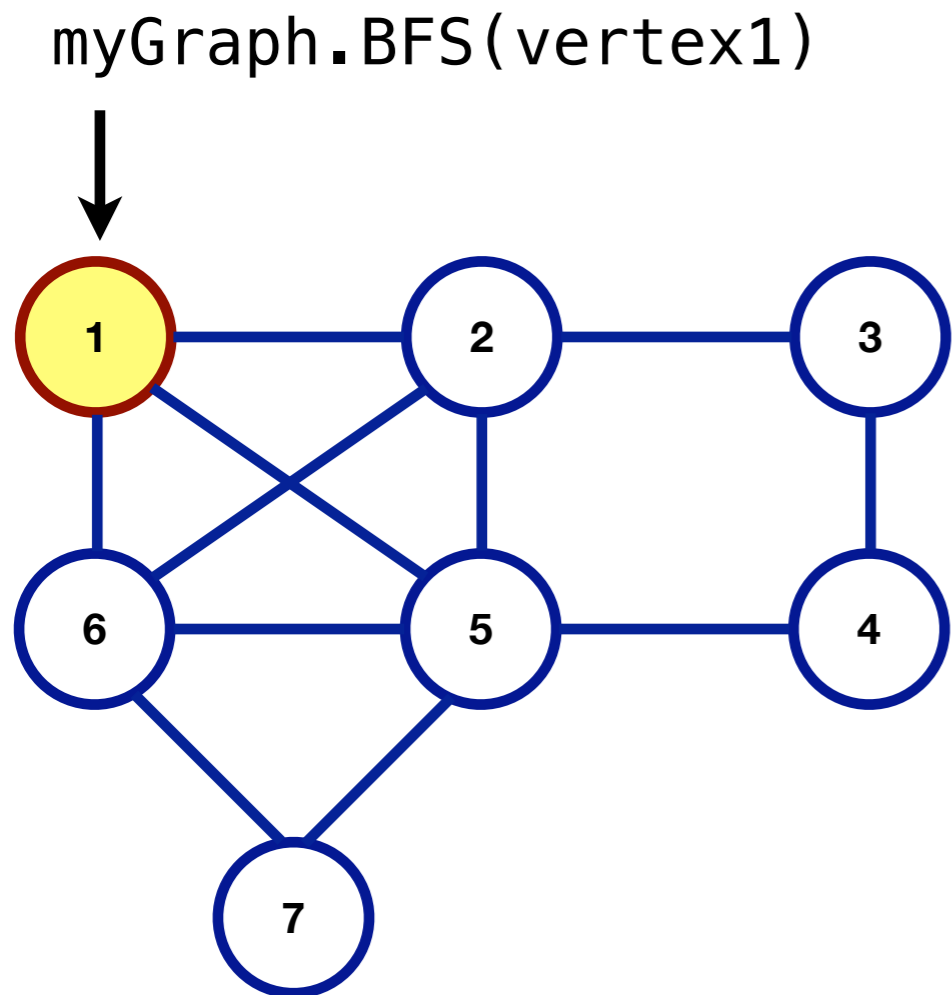
# Breadth-First Search / Traversal

---



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal



```
proc BFS(fromVertex v)
```

```
  Q := new Queue()
```

```
  Q.enqueue(v)
```

```
  v.processed := true
```

```
  while (not Q.empty())
```

```
    cv := Q.dequeue()
```

```
    print(cv.id)
```

```
    for n in cv.neighbors[]
```

```
      if (not n.processed)
```

```
        Q.enqueue(n)
```

```
        n.processed := true
```

```
      endif
```

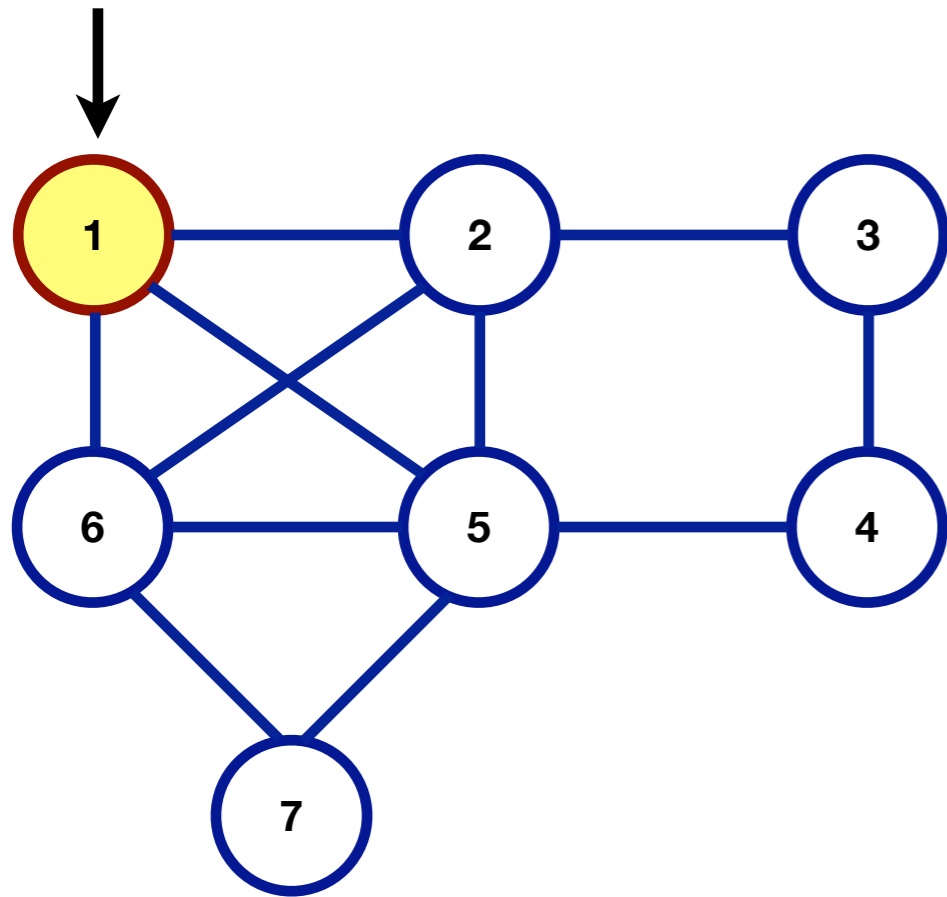
```
    endfor
```

```
  endwhile
```

```
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



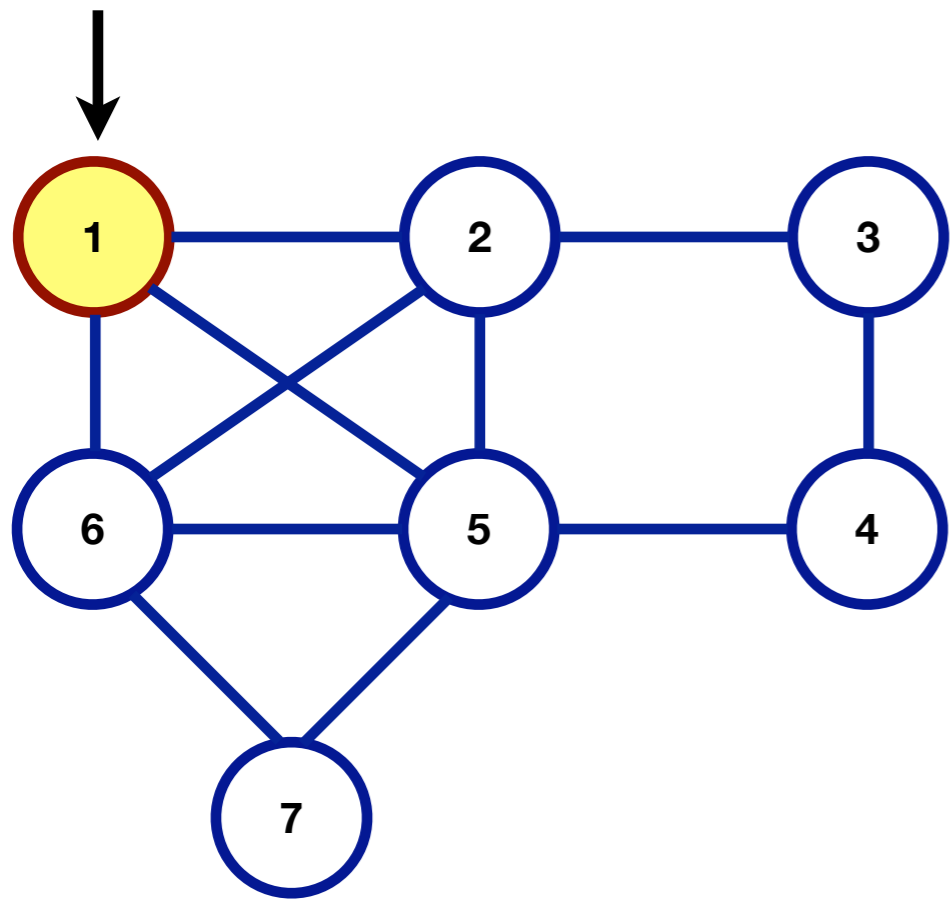
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```





# Breadth-First Search / Traversal

myGraph.BFS(vertex1)

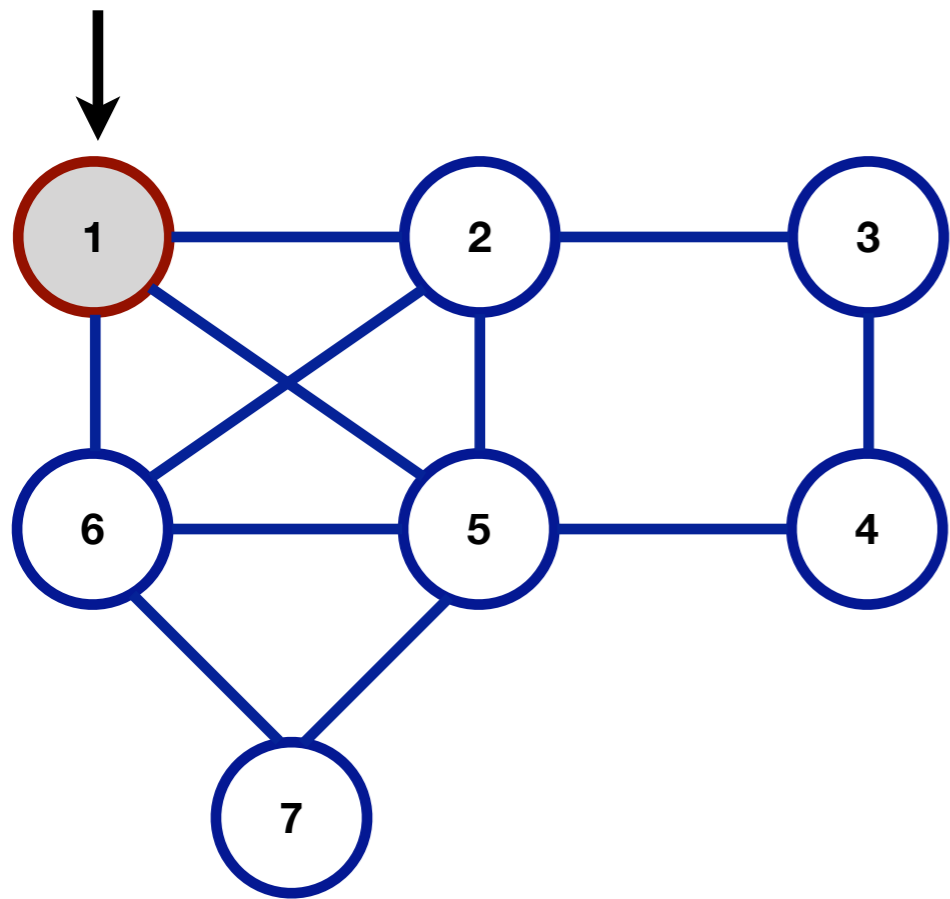


```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)

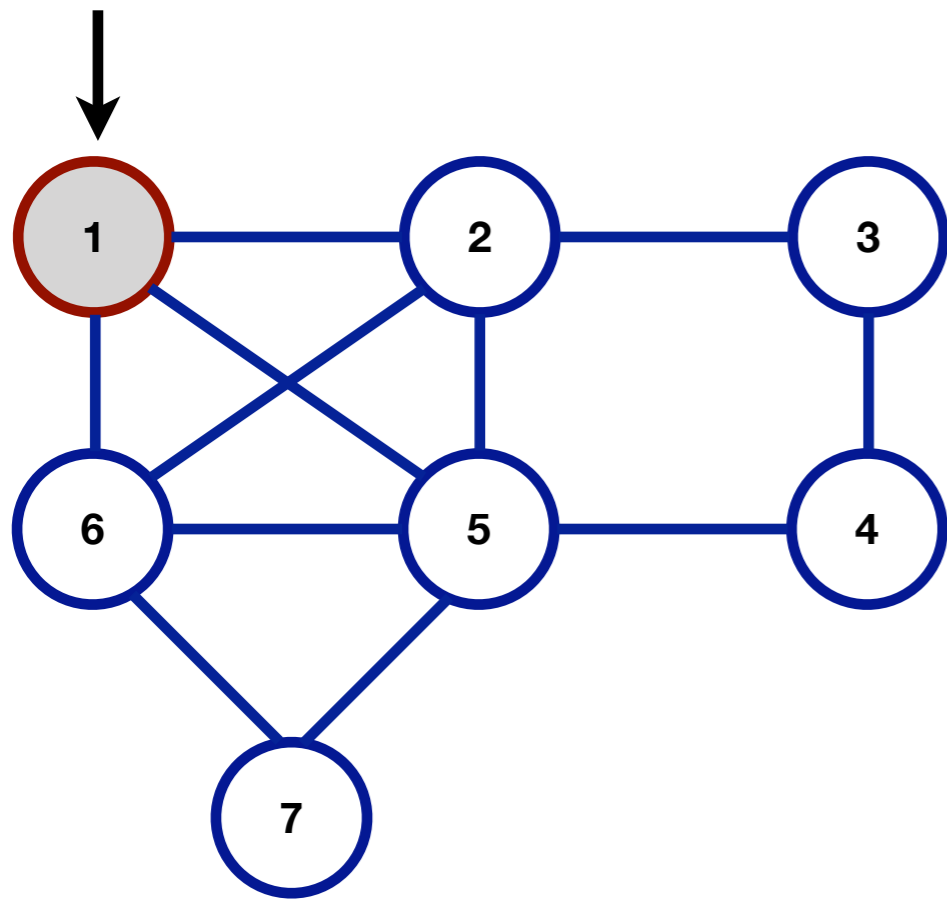


```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```



# Breadth-First Search / Traversal

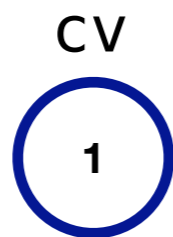
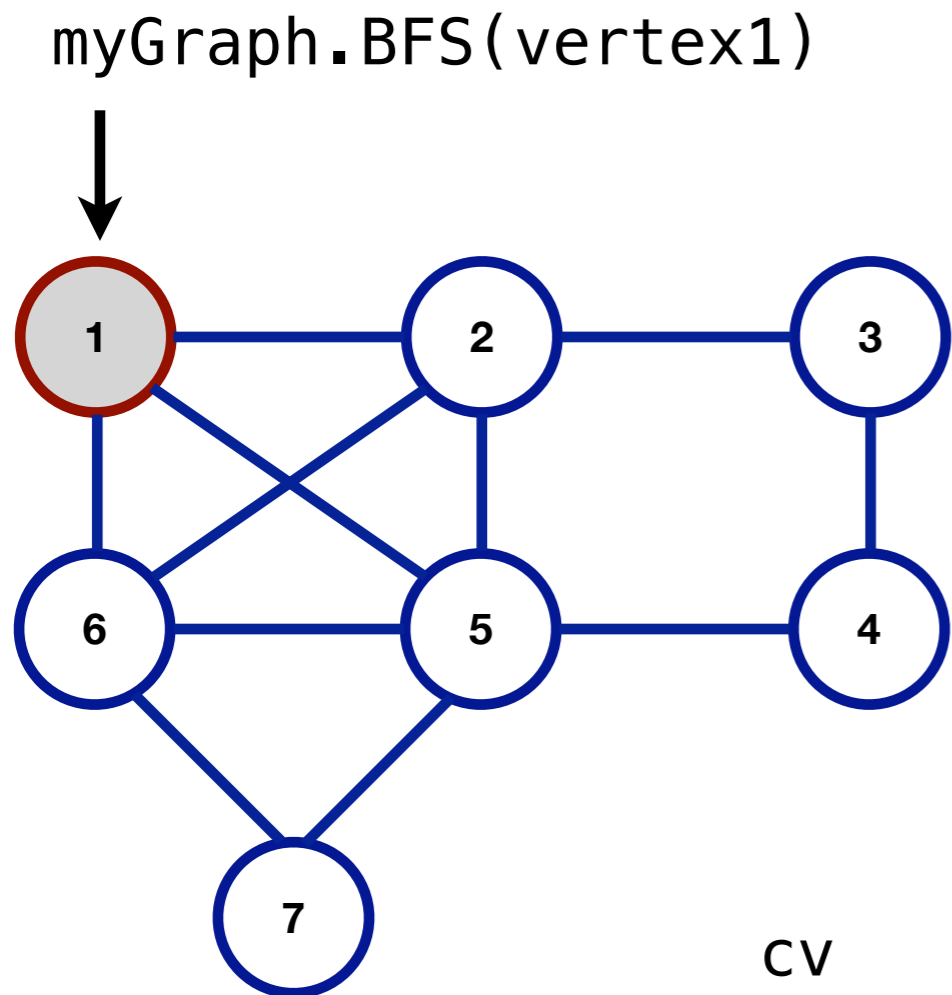
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

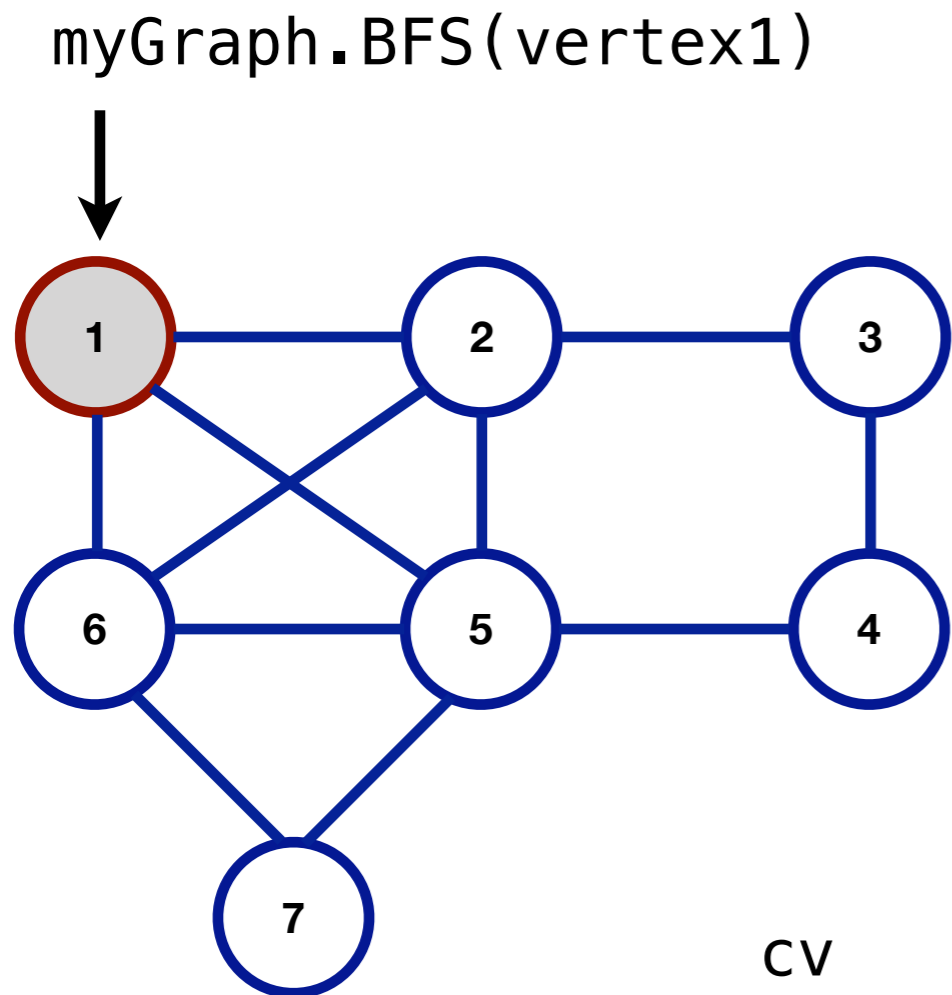


# Breadth-First Search / Traversal



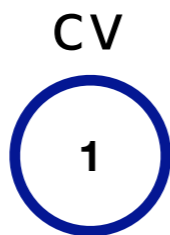
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal



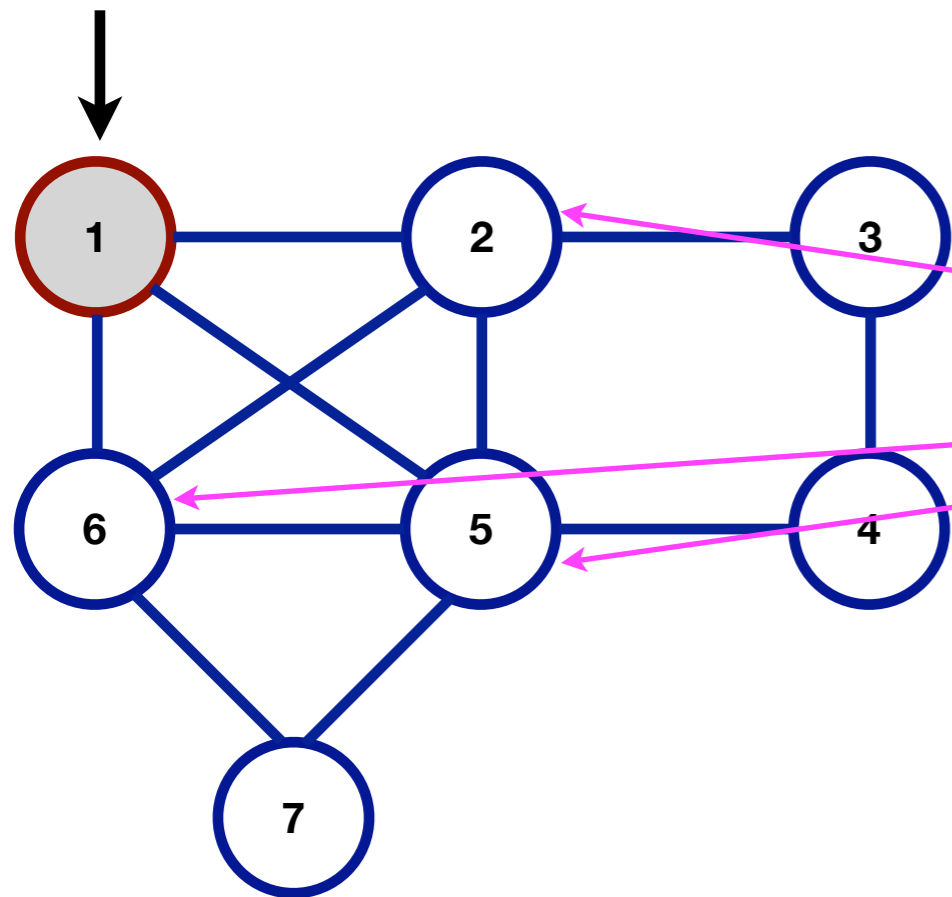
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

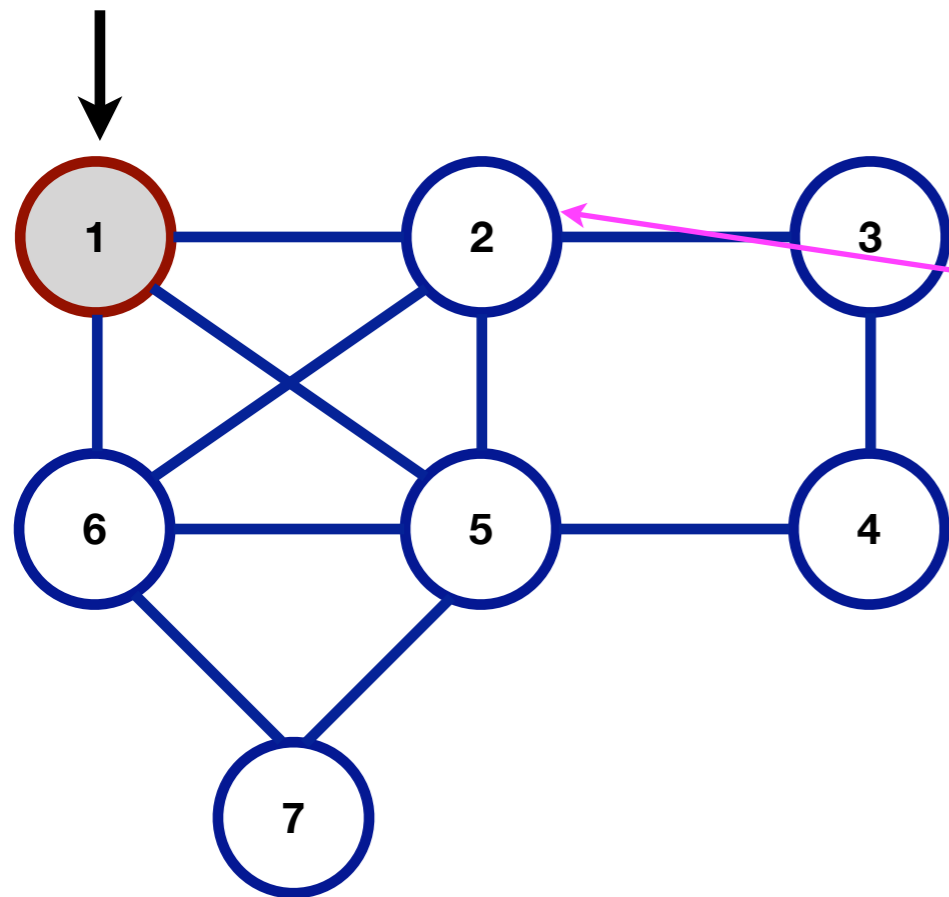
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1



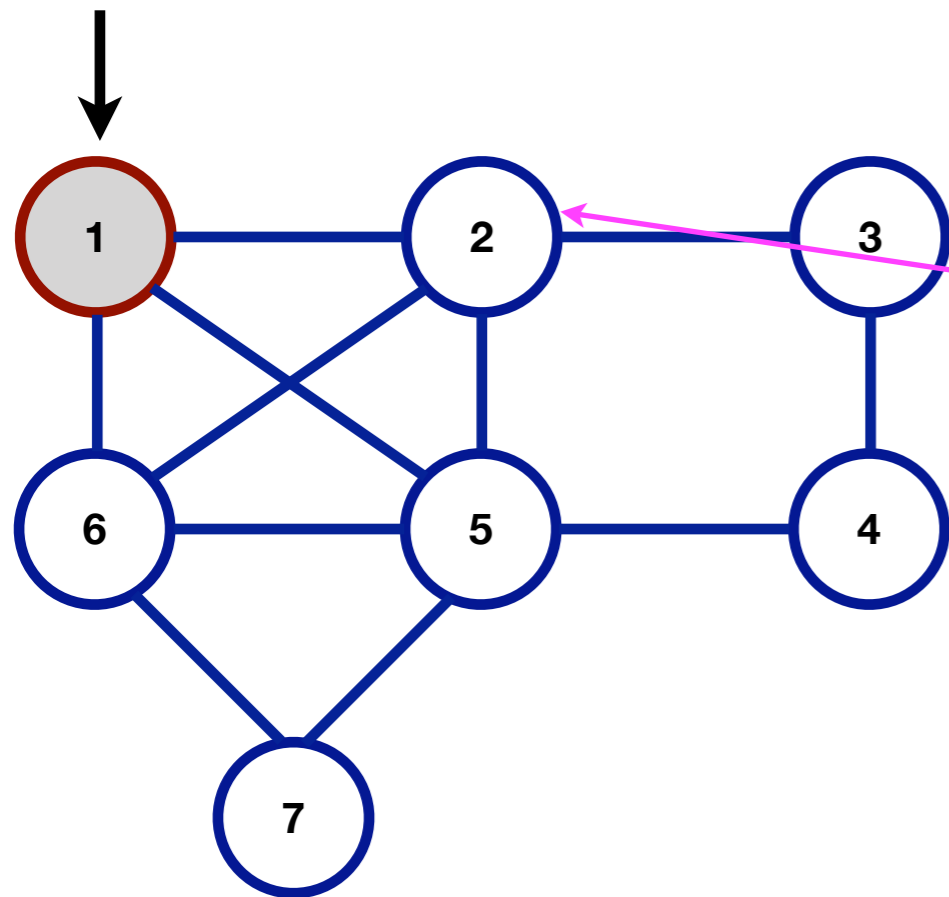
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

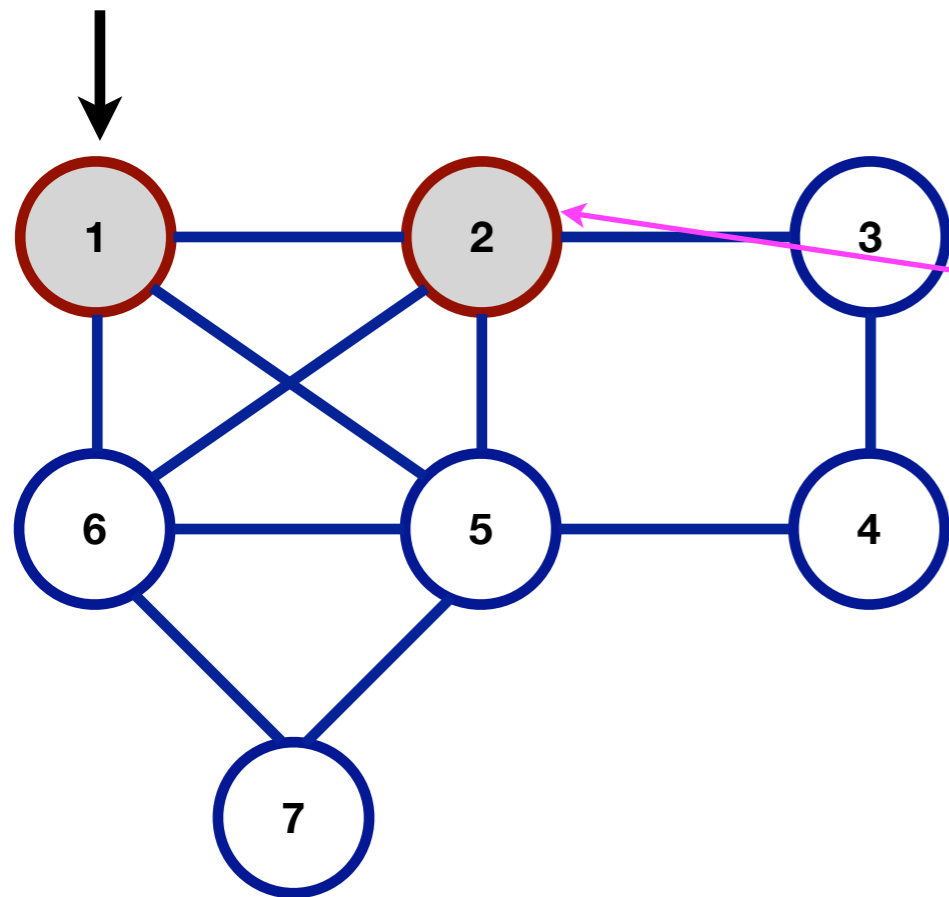
1





# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

n.processed := true

1

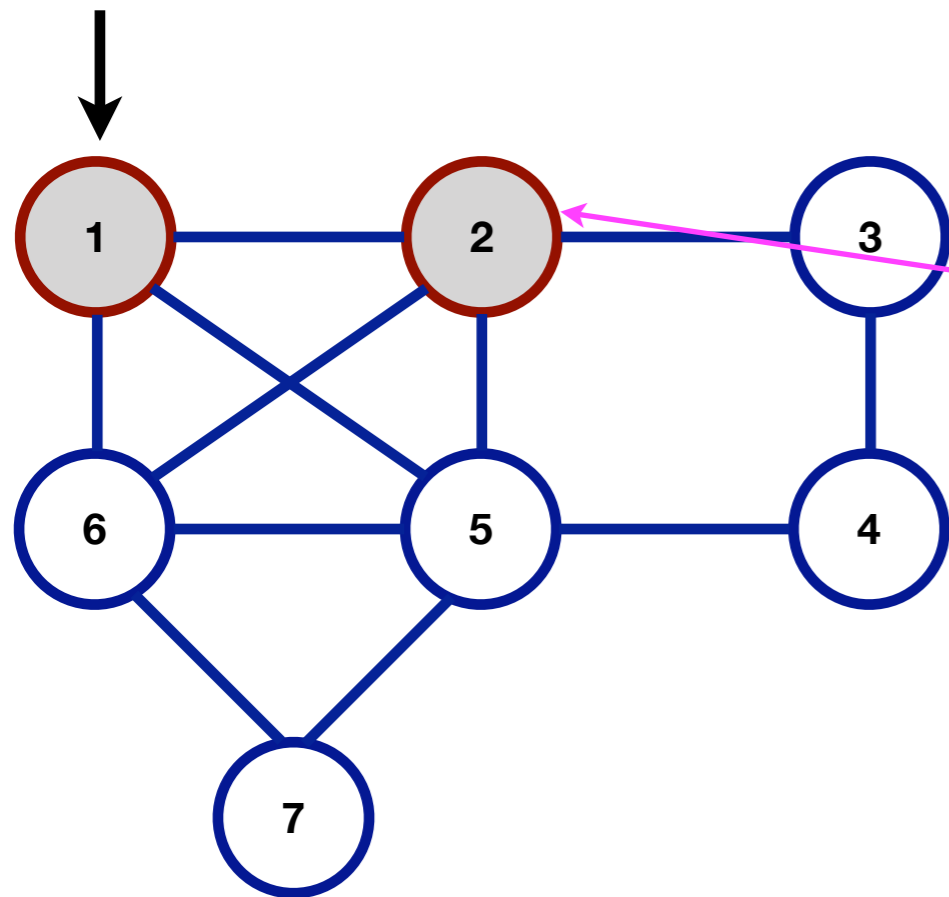


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endif

endfor

endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

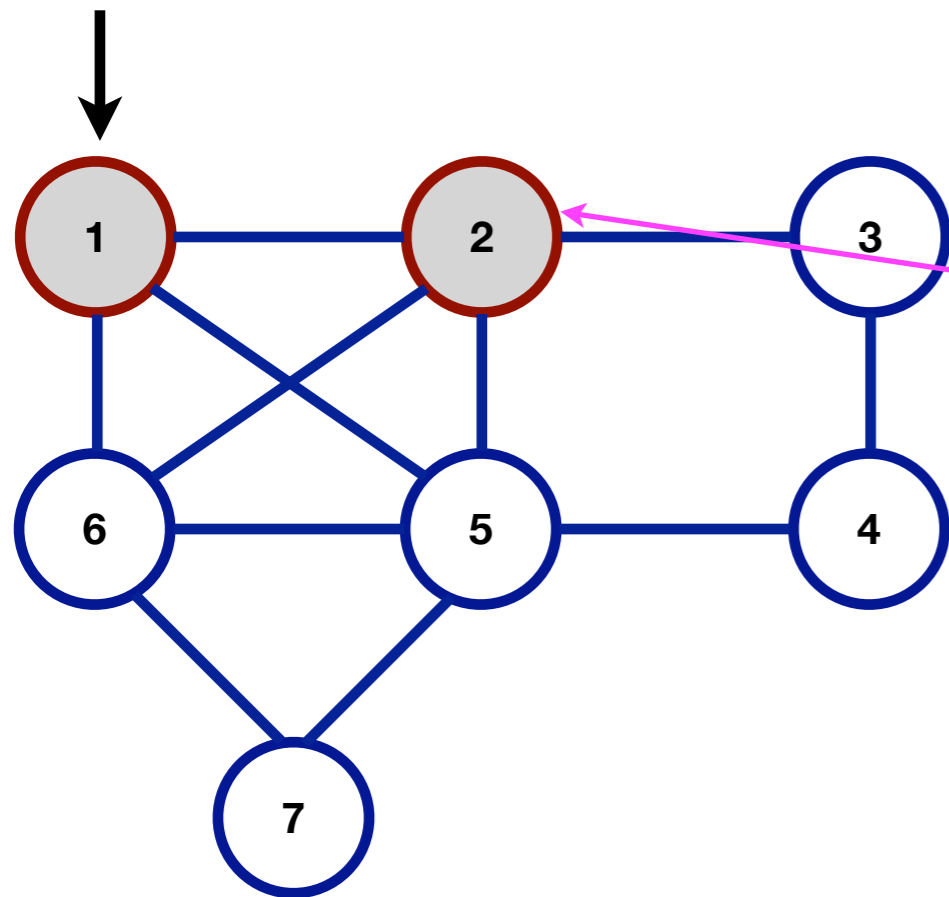
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1

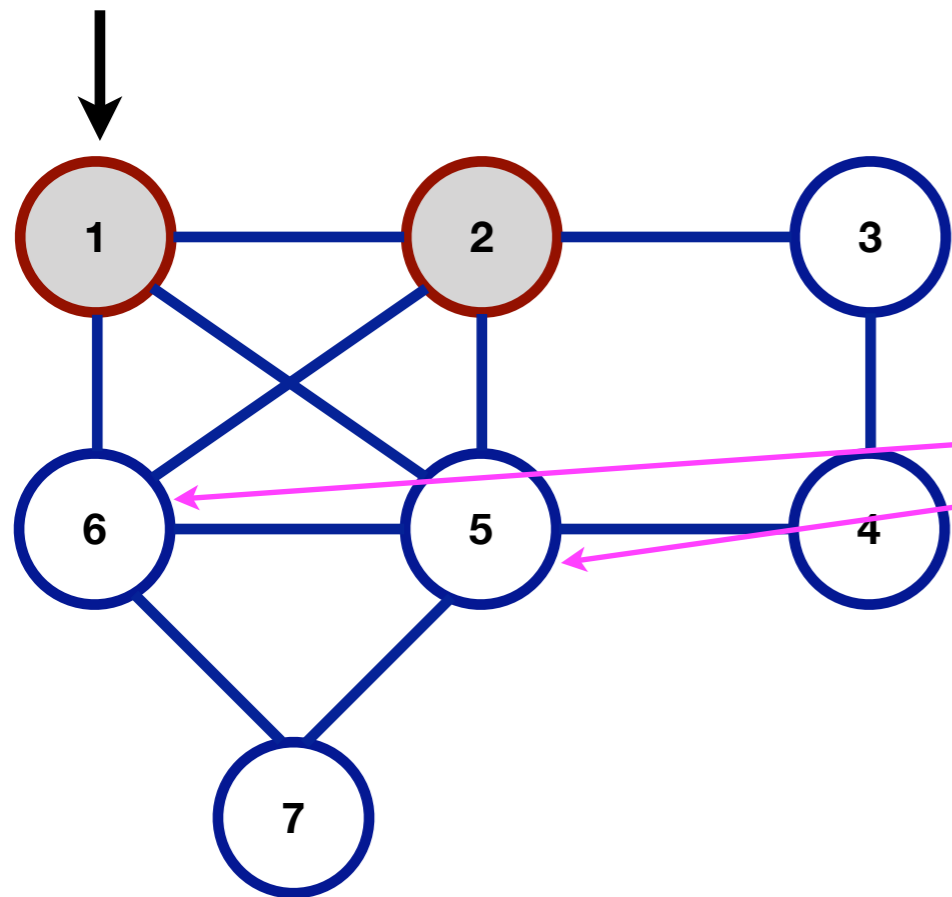


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

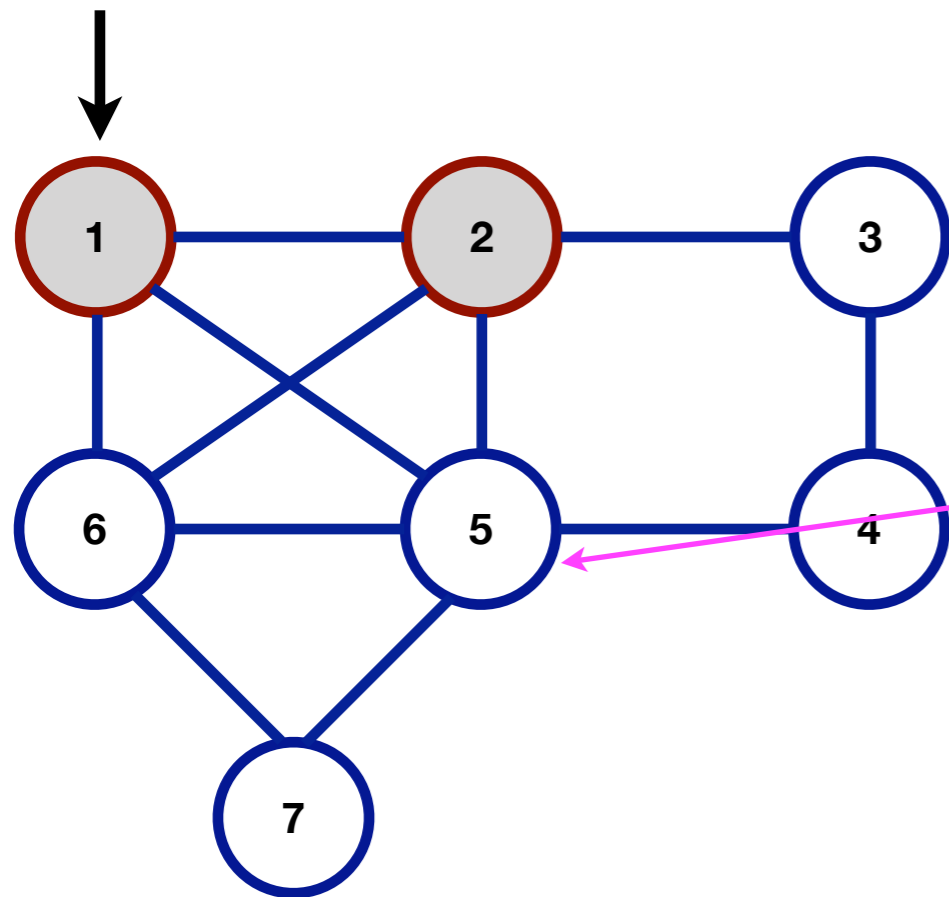
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

for n in cv.neighbors[]

if (not n.processed)

Q.enqueue(n)

n.processed := true

endif

endfor

endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

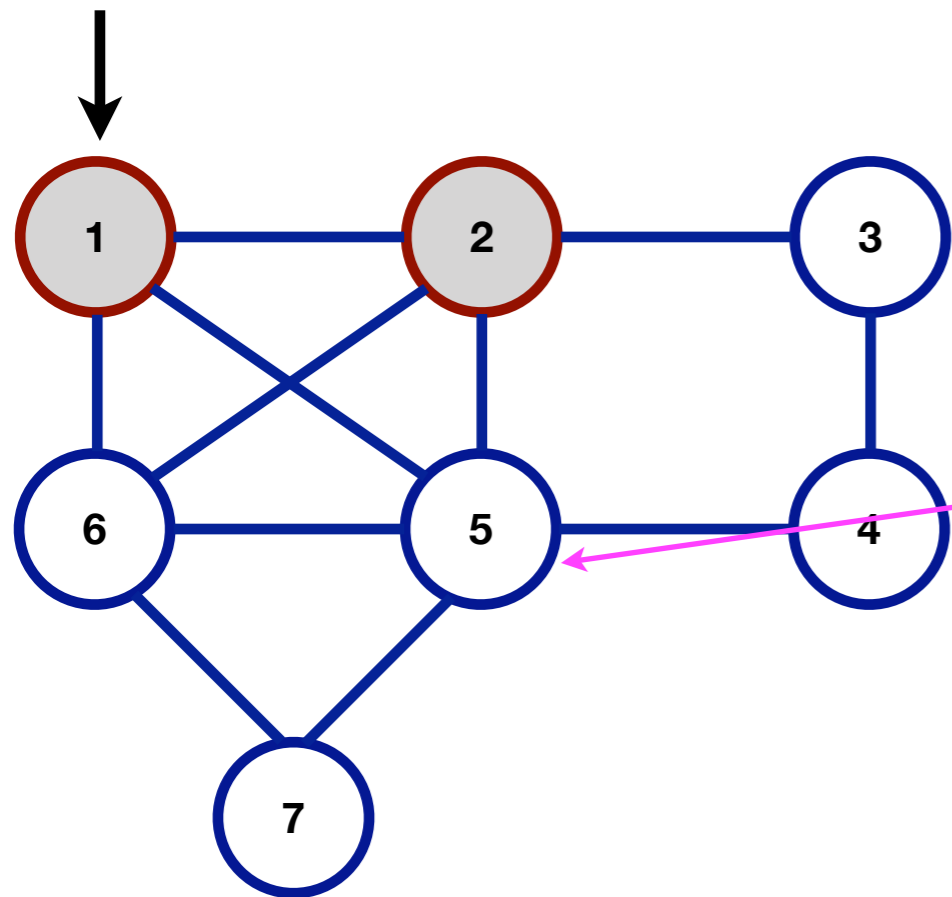
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1

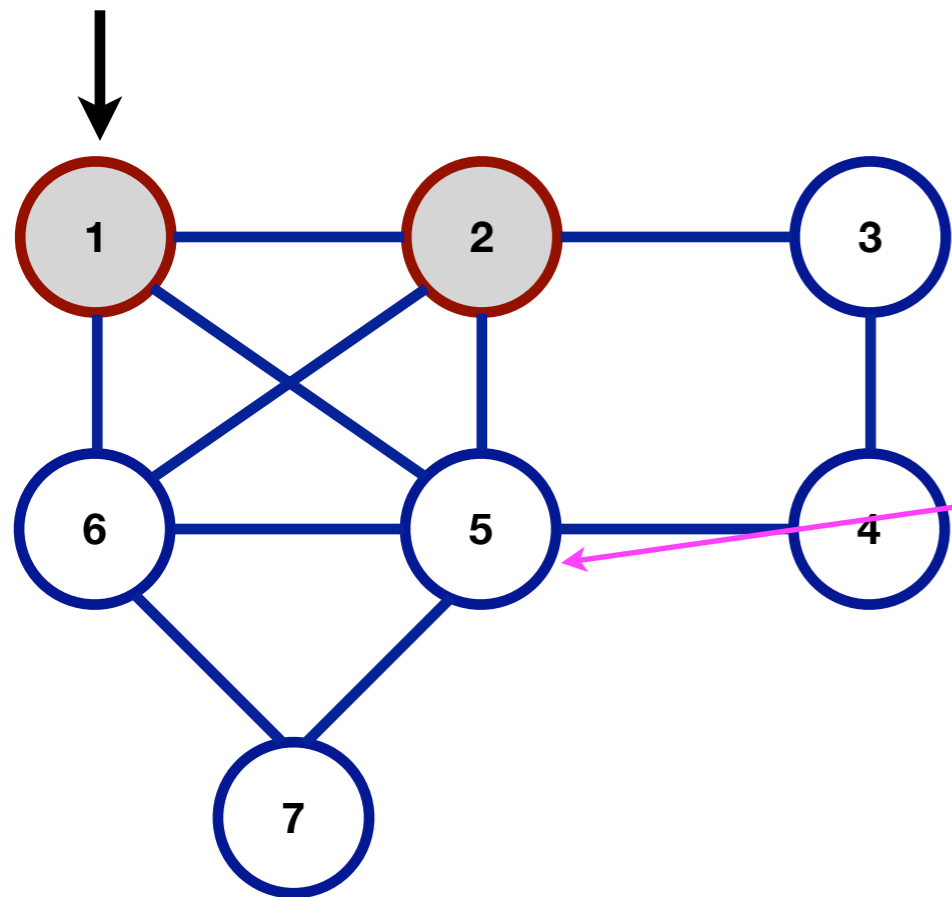


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1

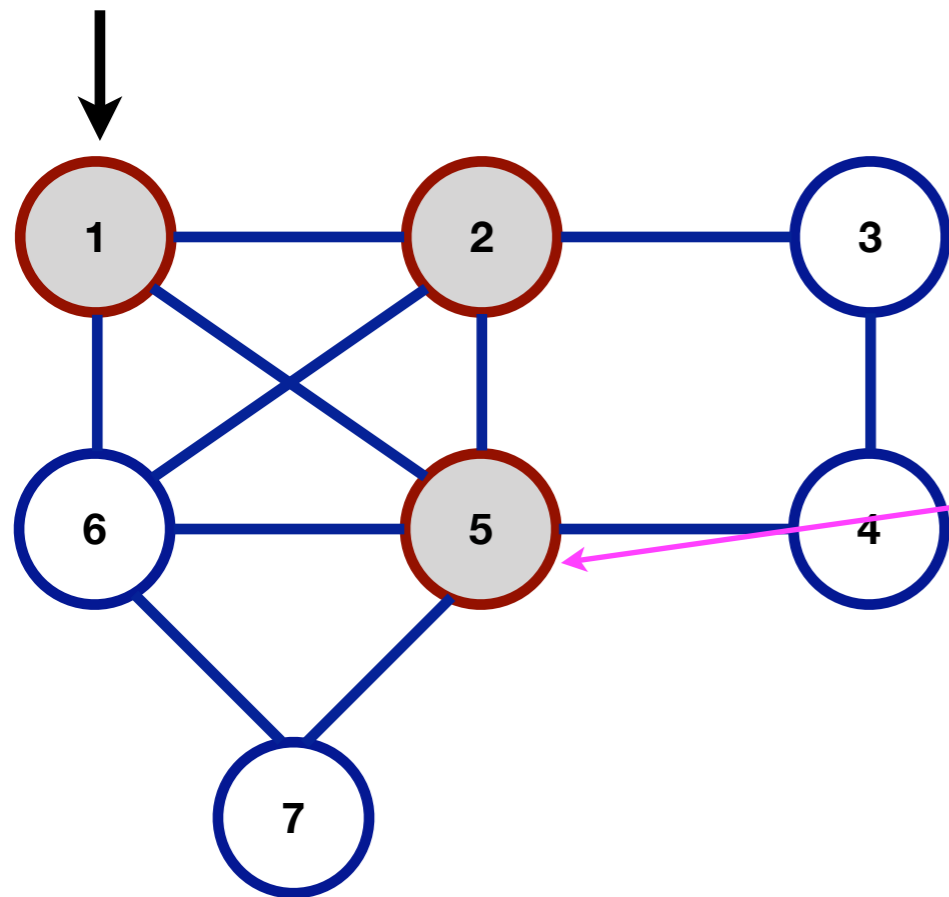


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

n.processed := true

1



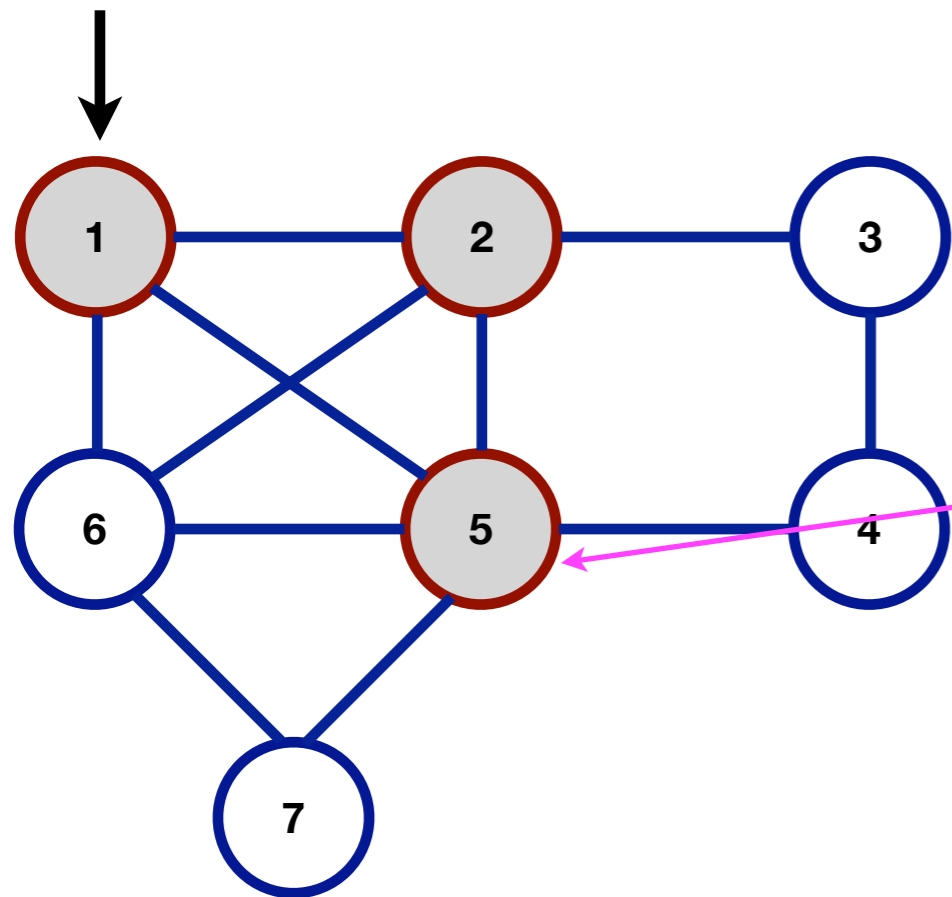
```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endif

endfor

endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

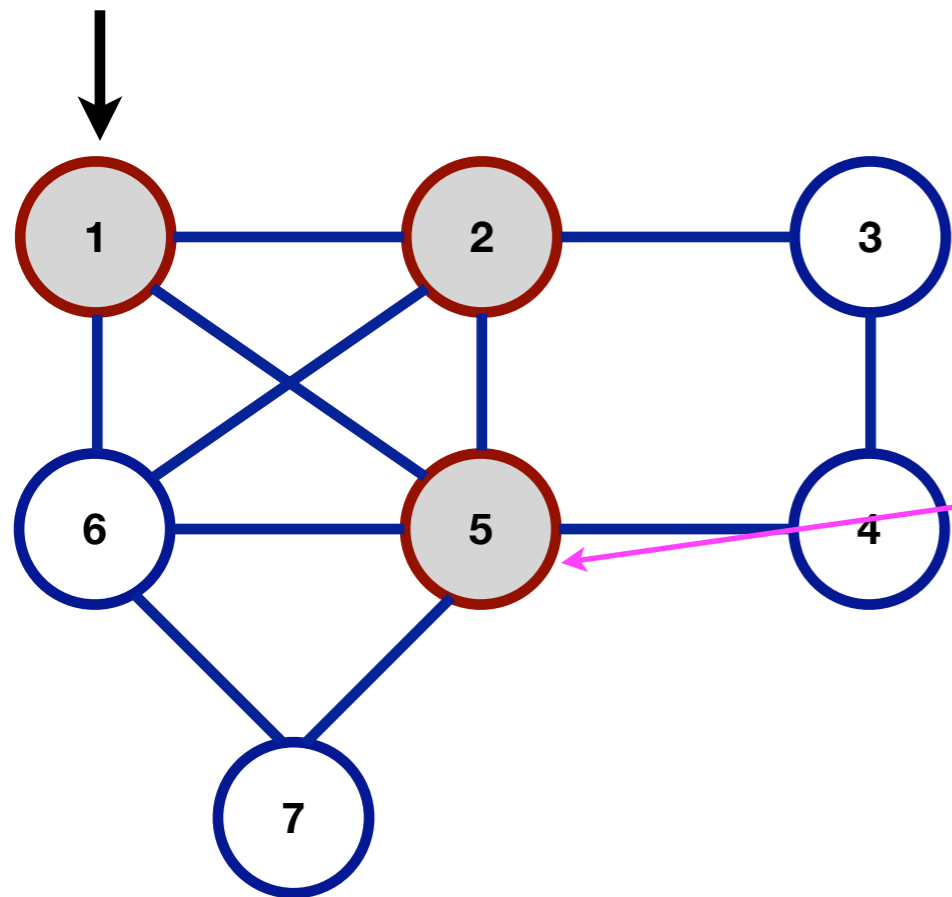
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1

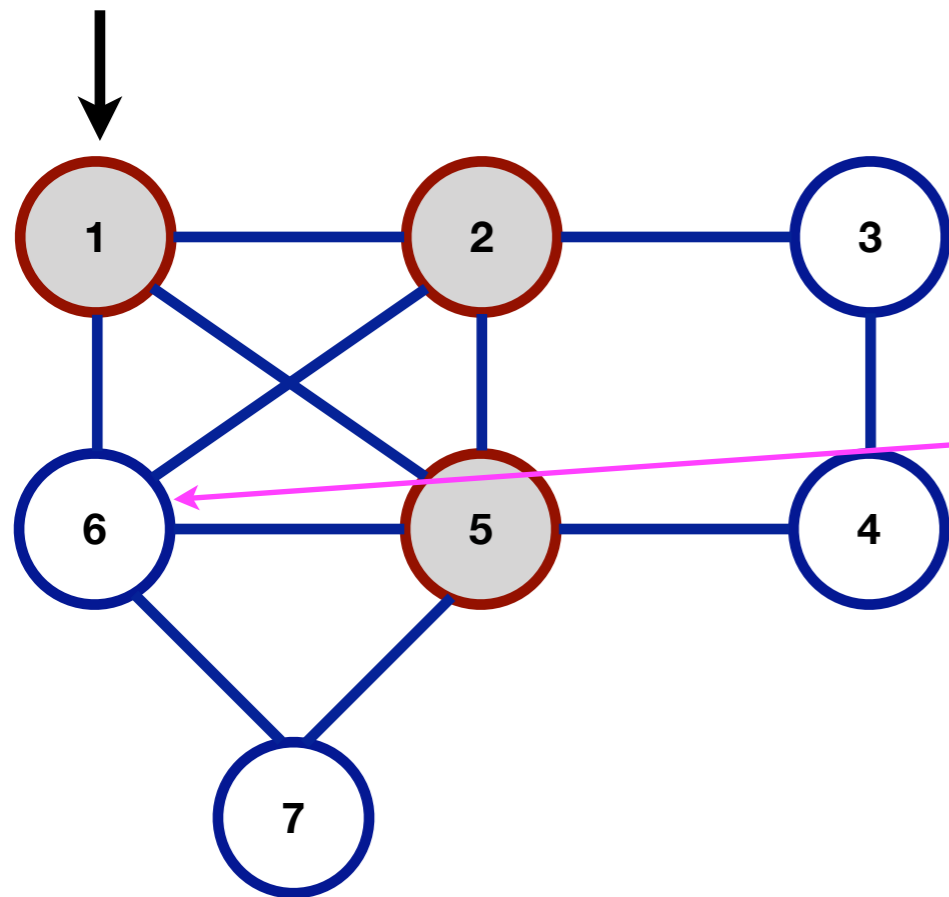


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

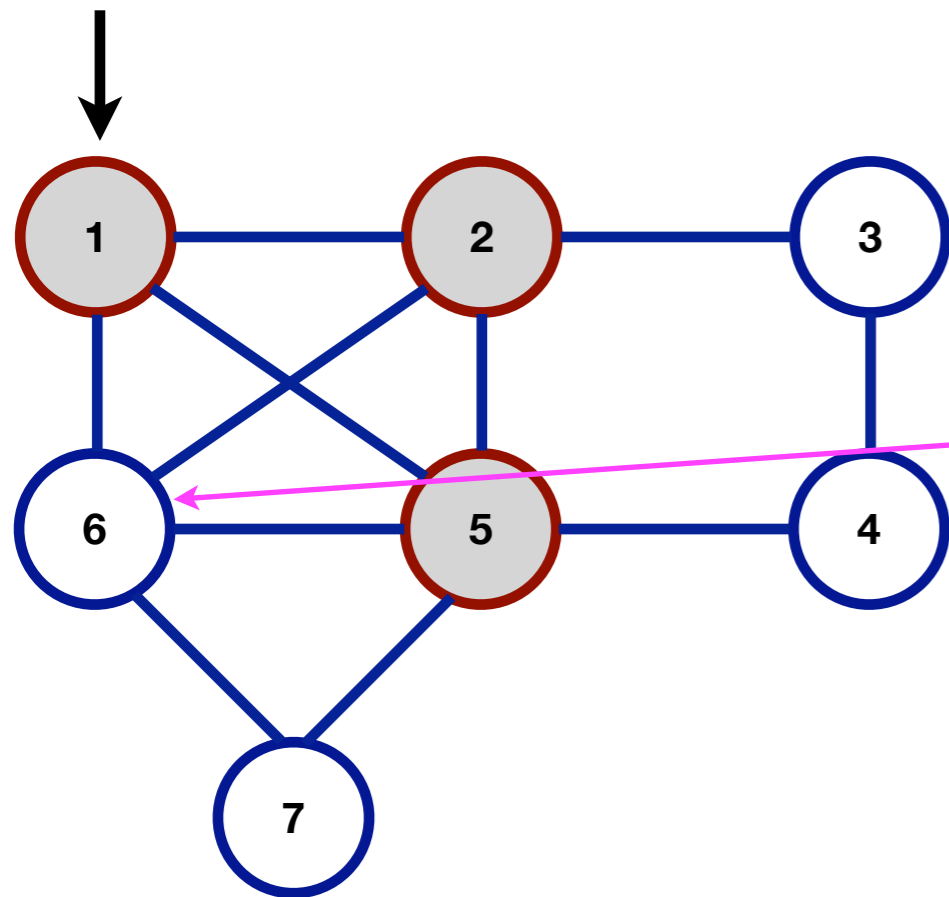
Adjacency List

1



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1

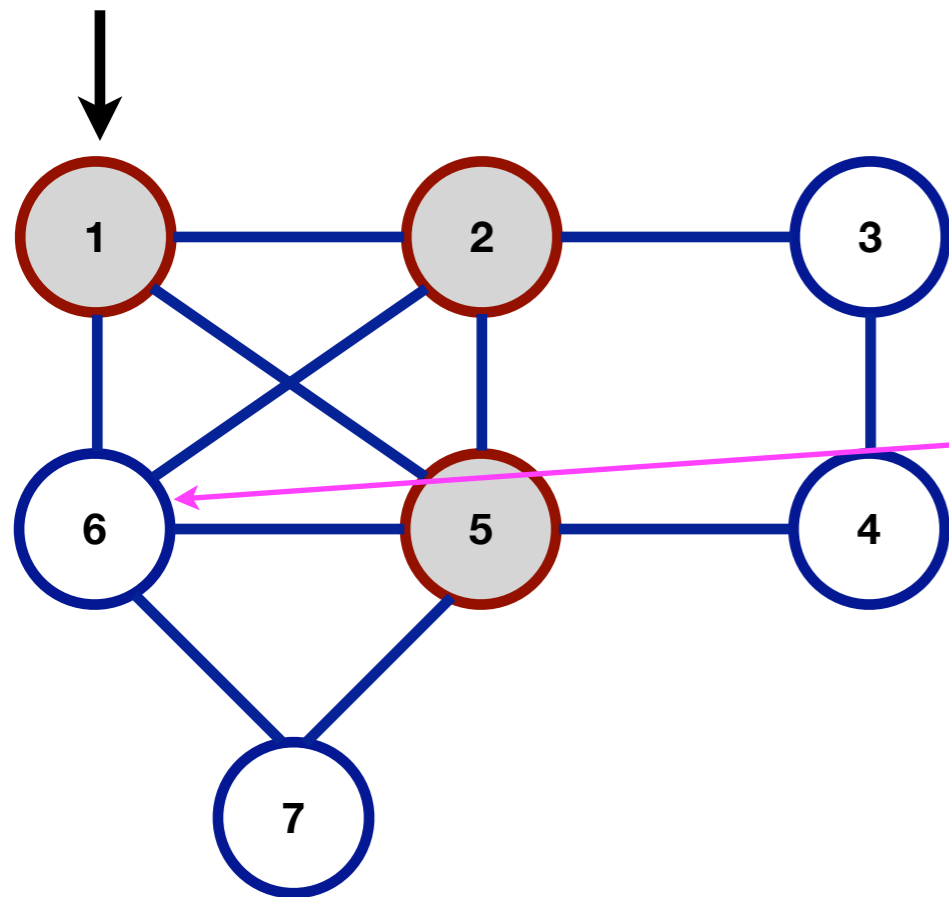


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

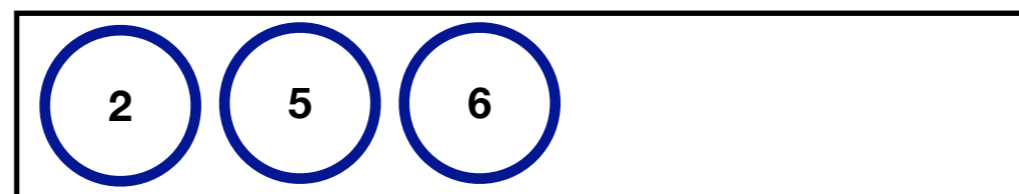
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

Q.enqueue(n)

1

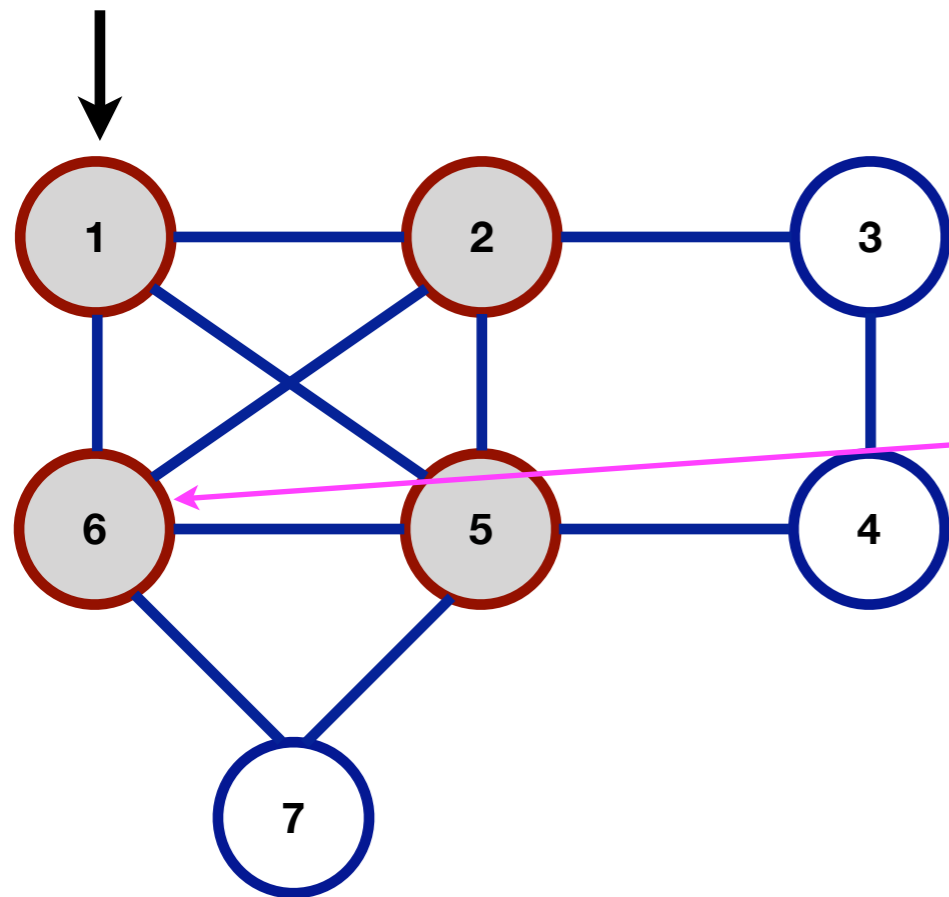


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

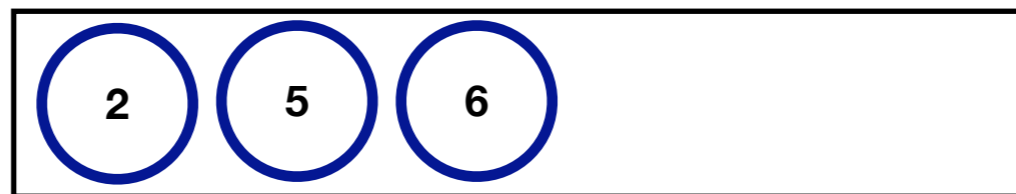
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

n.processed := true

1

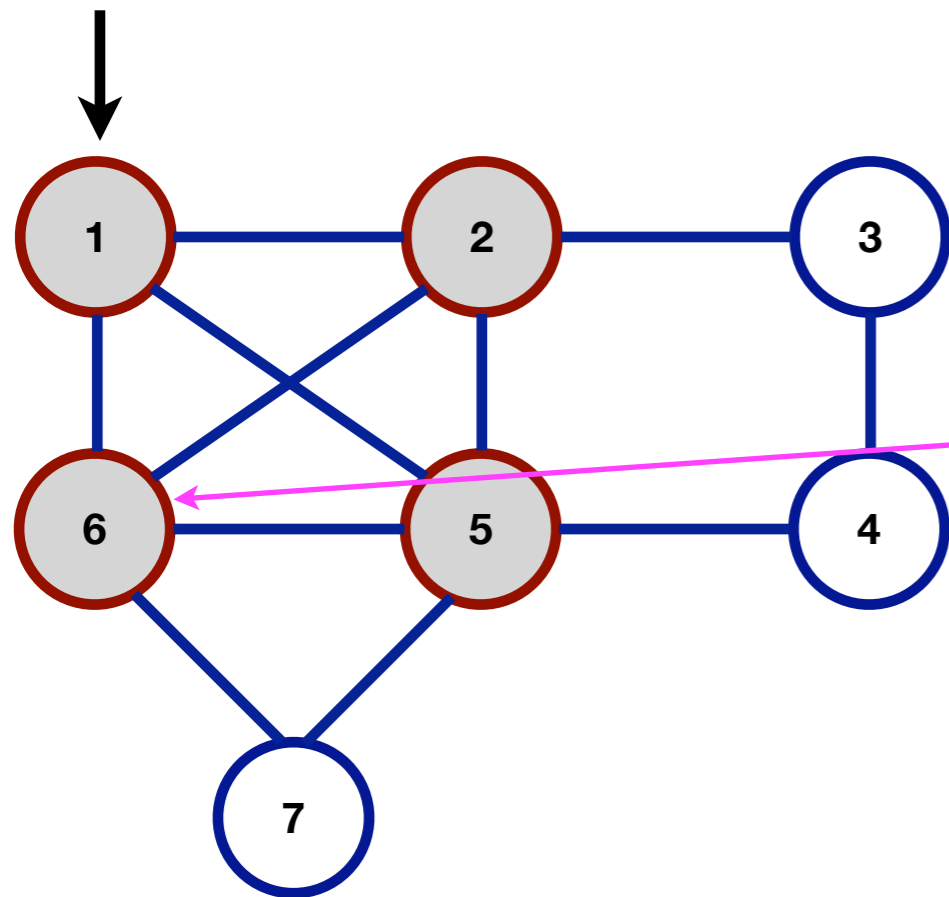


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endif

endfor

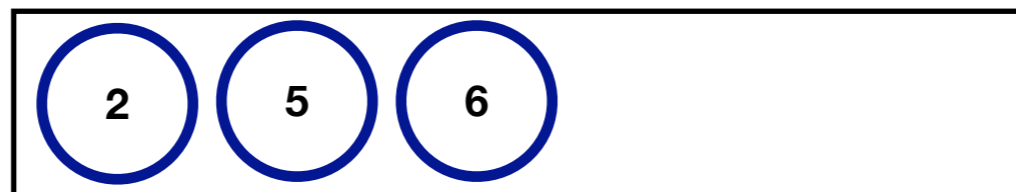
endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

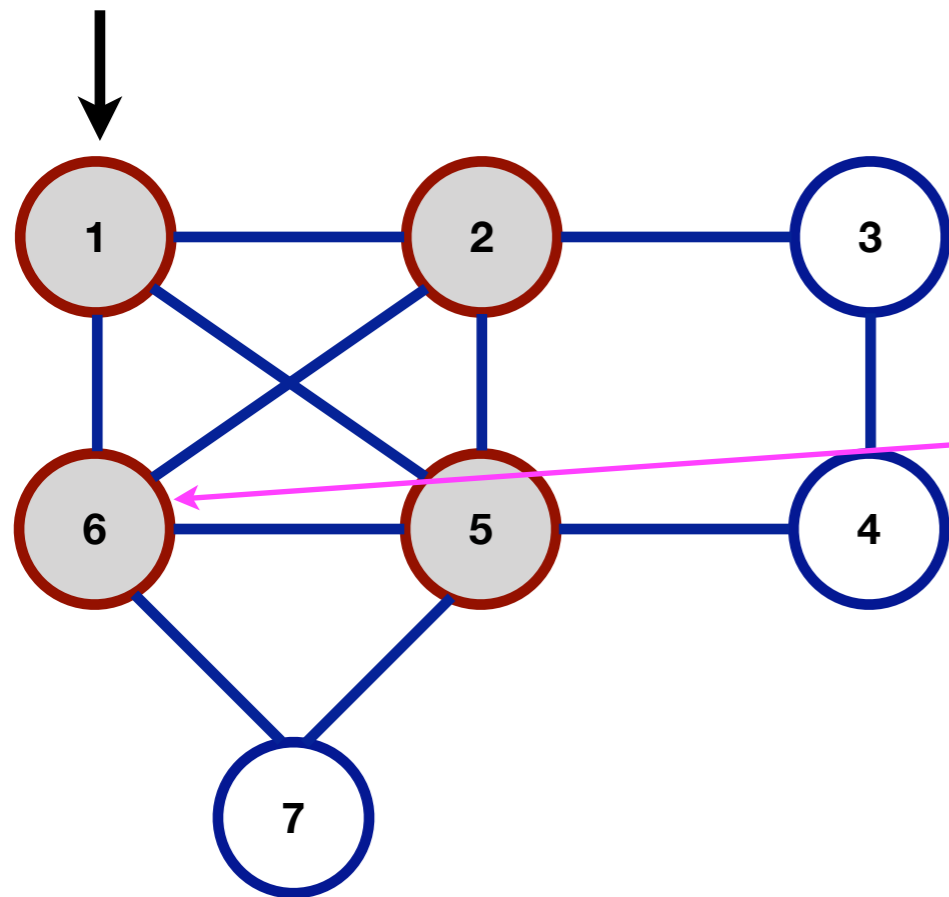
Adjacency List

1



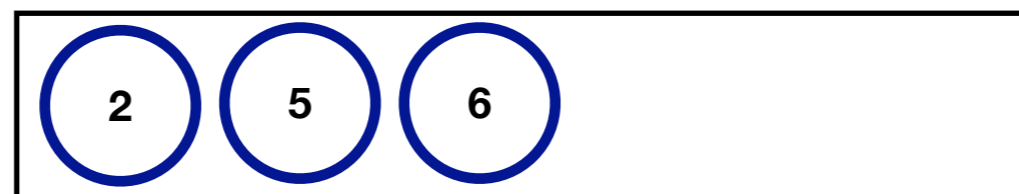
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1



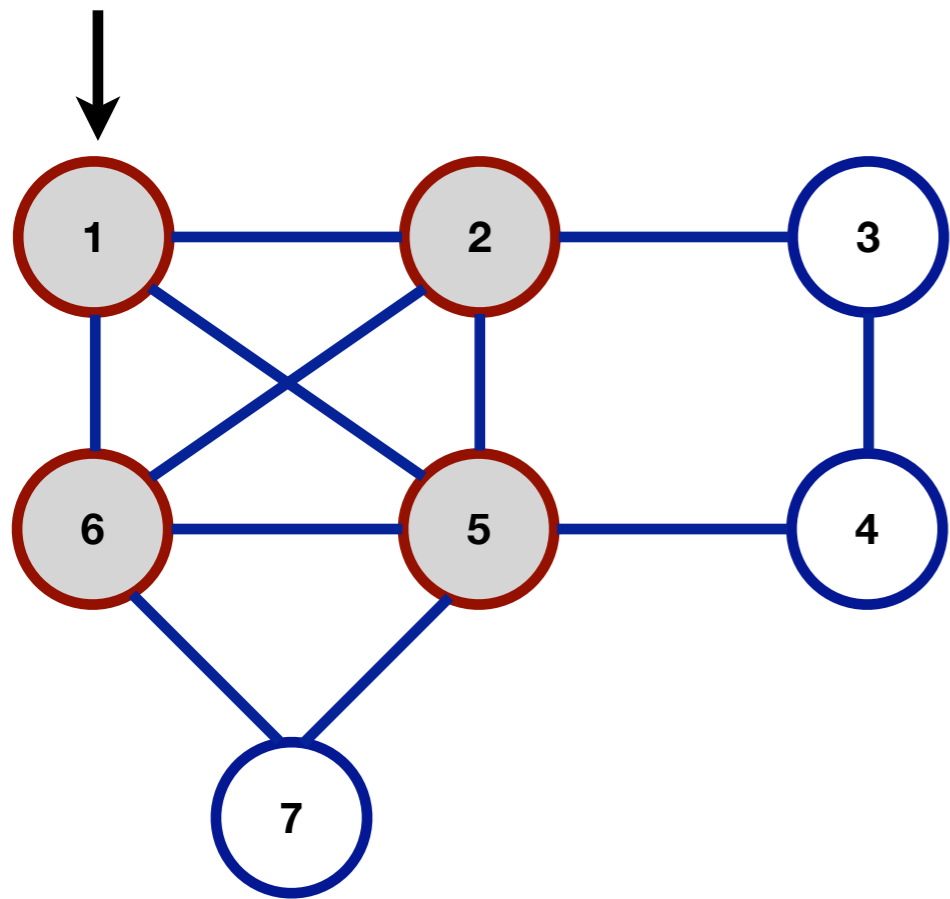
```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List



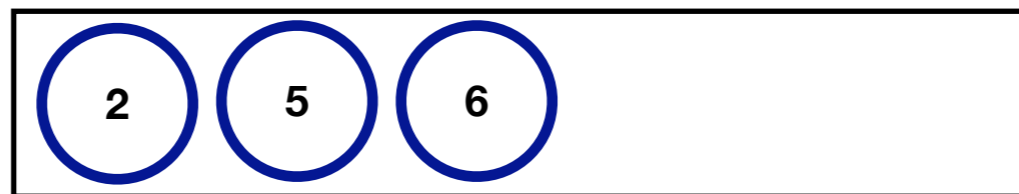
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



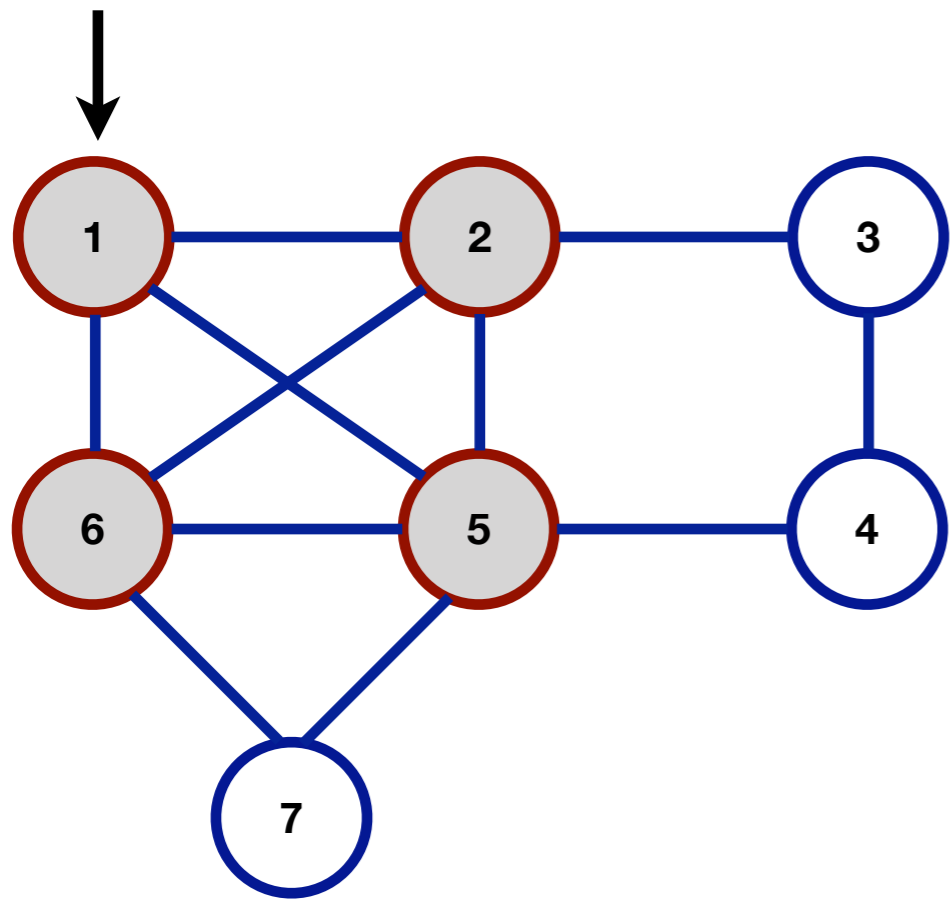
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1



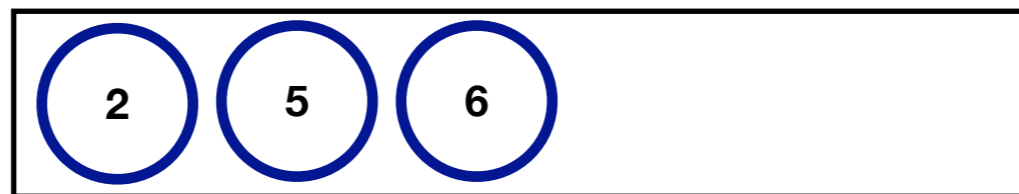
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



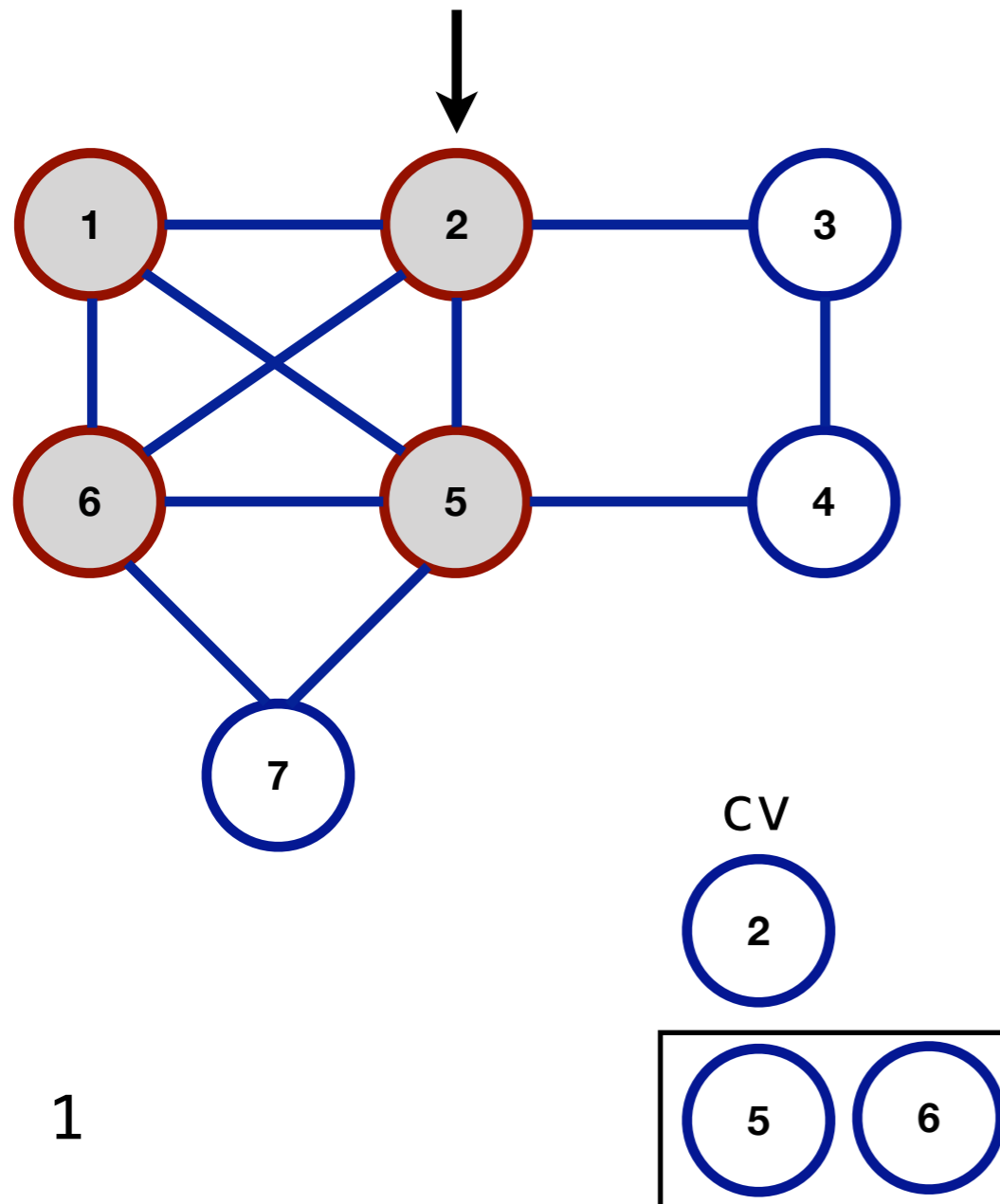
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1



# Breadth-First Search / Traversal

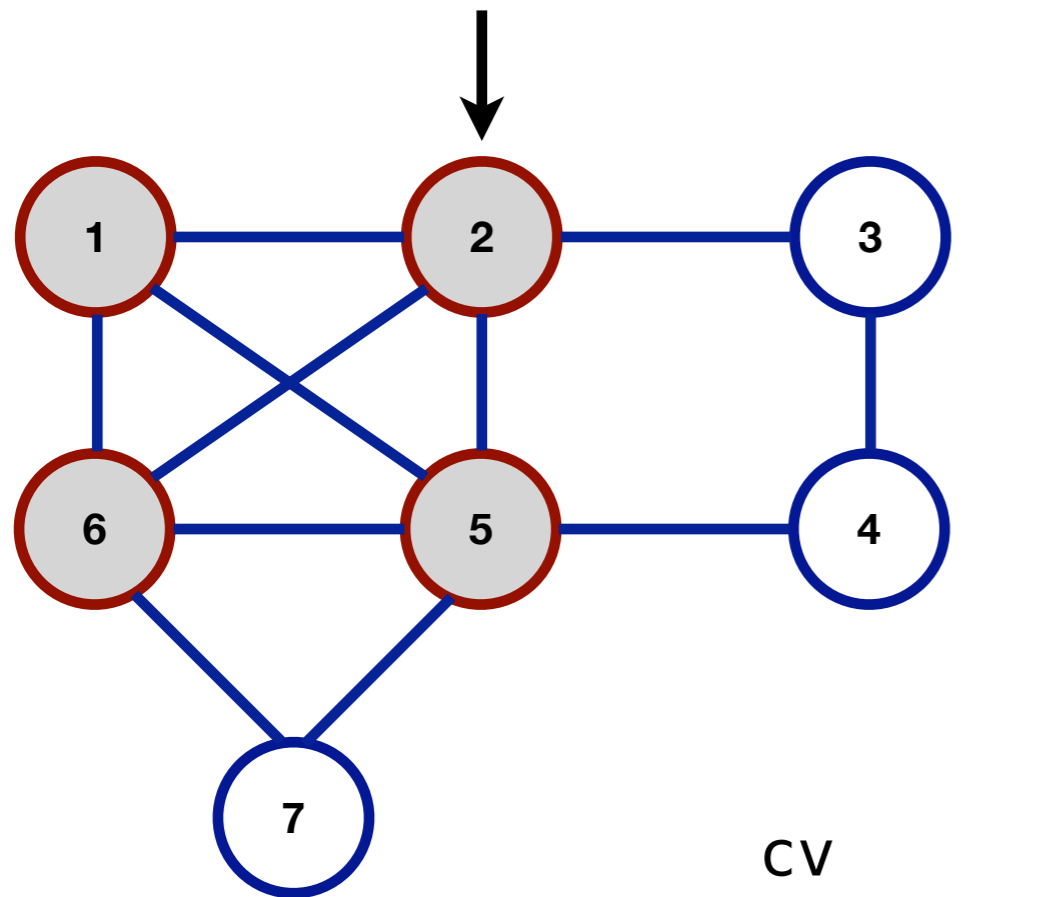
myGraph.BFS(vertex1)



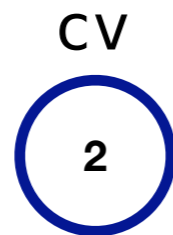
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



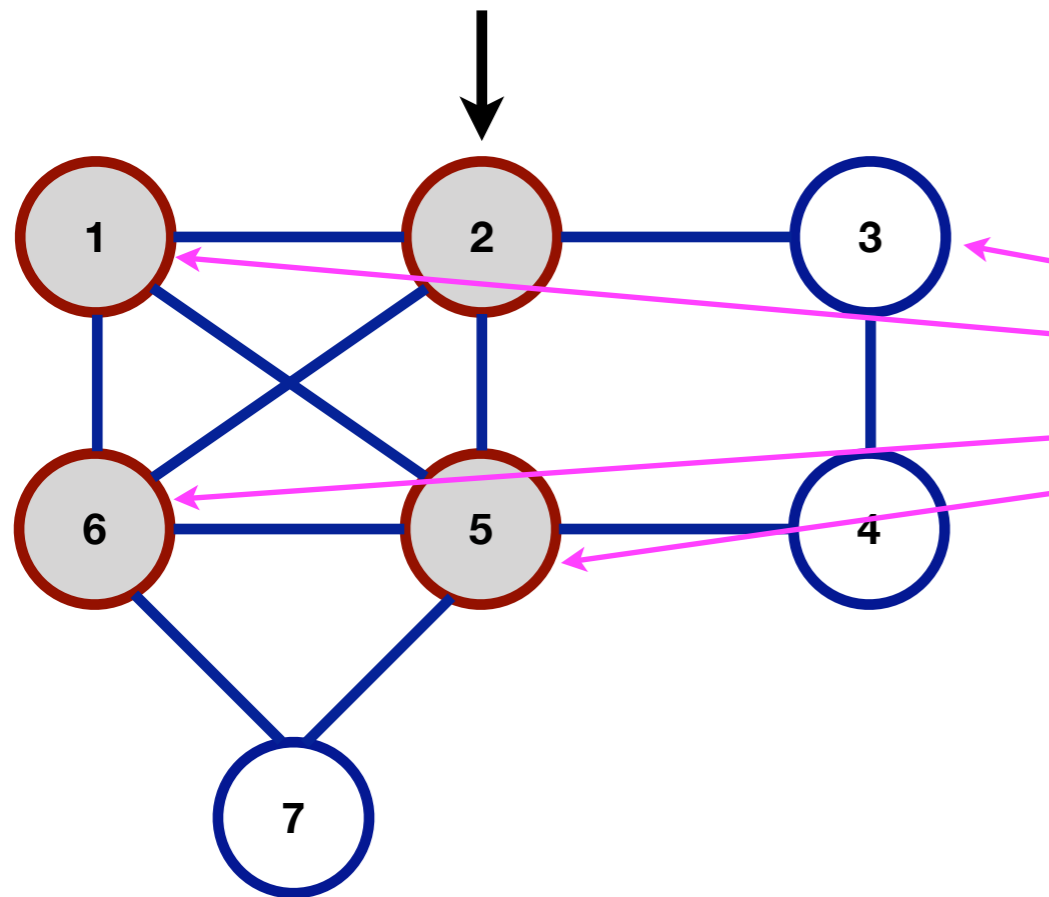
1 2



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

for n in cv.neighbors[]

1 2

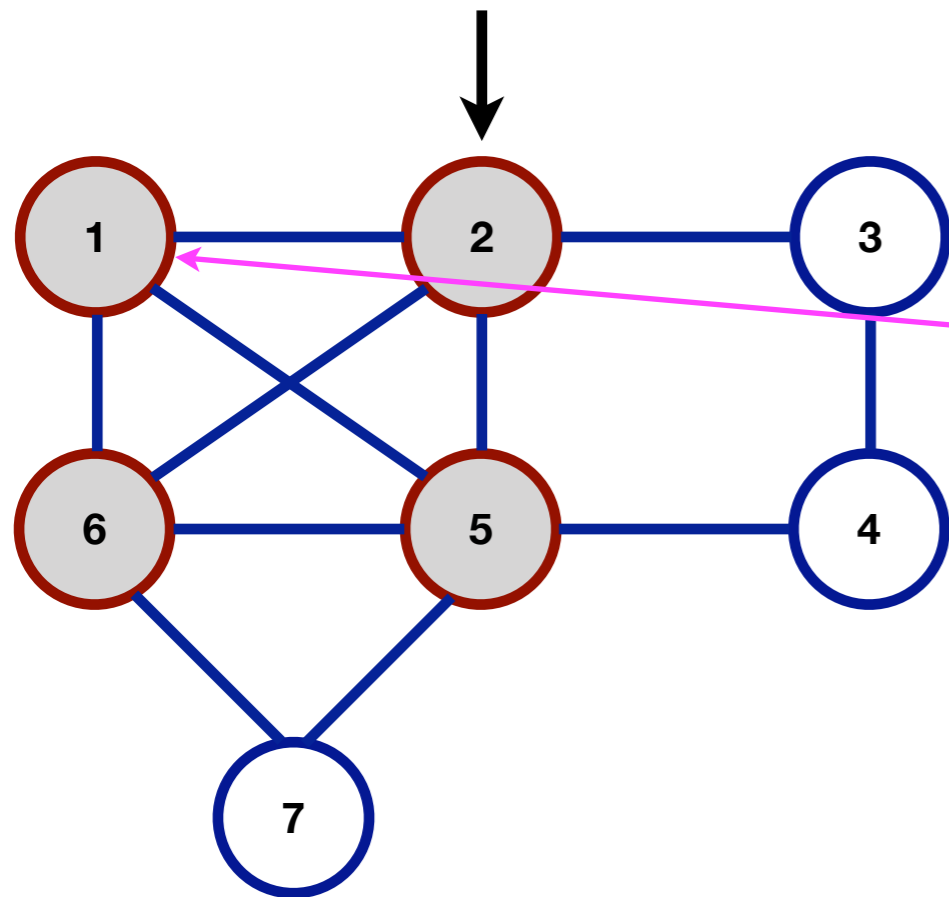


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2



```

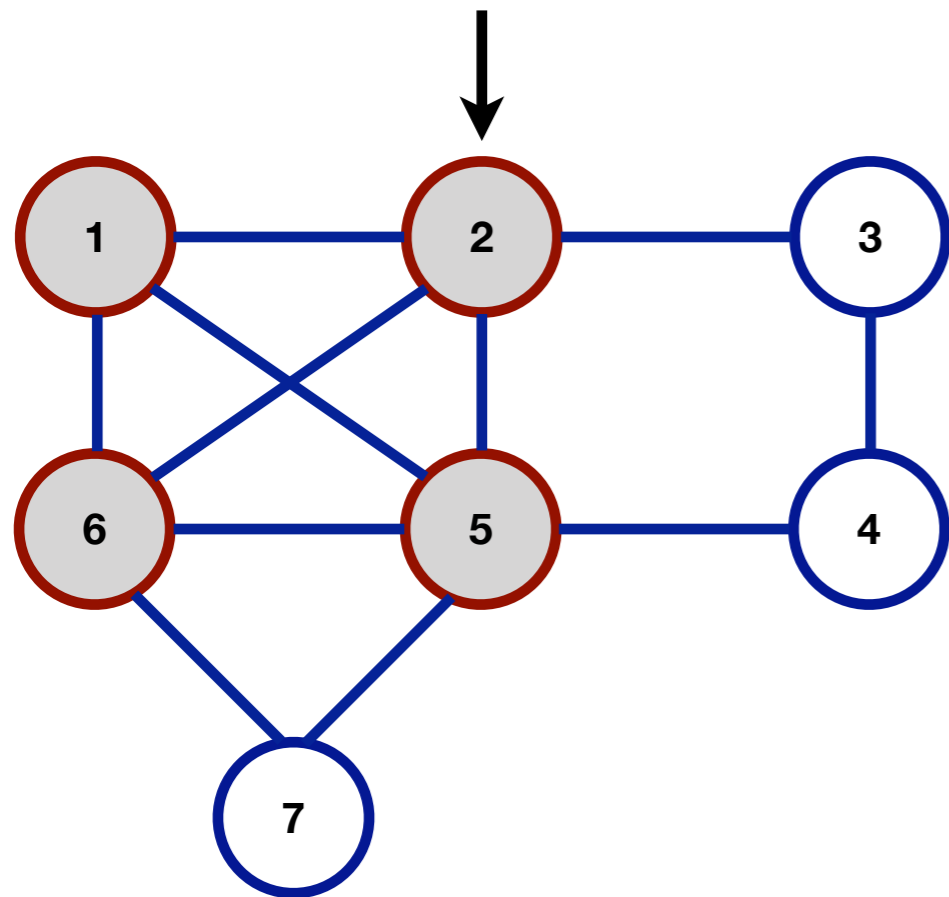
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
  
```

[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1 2

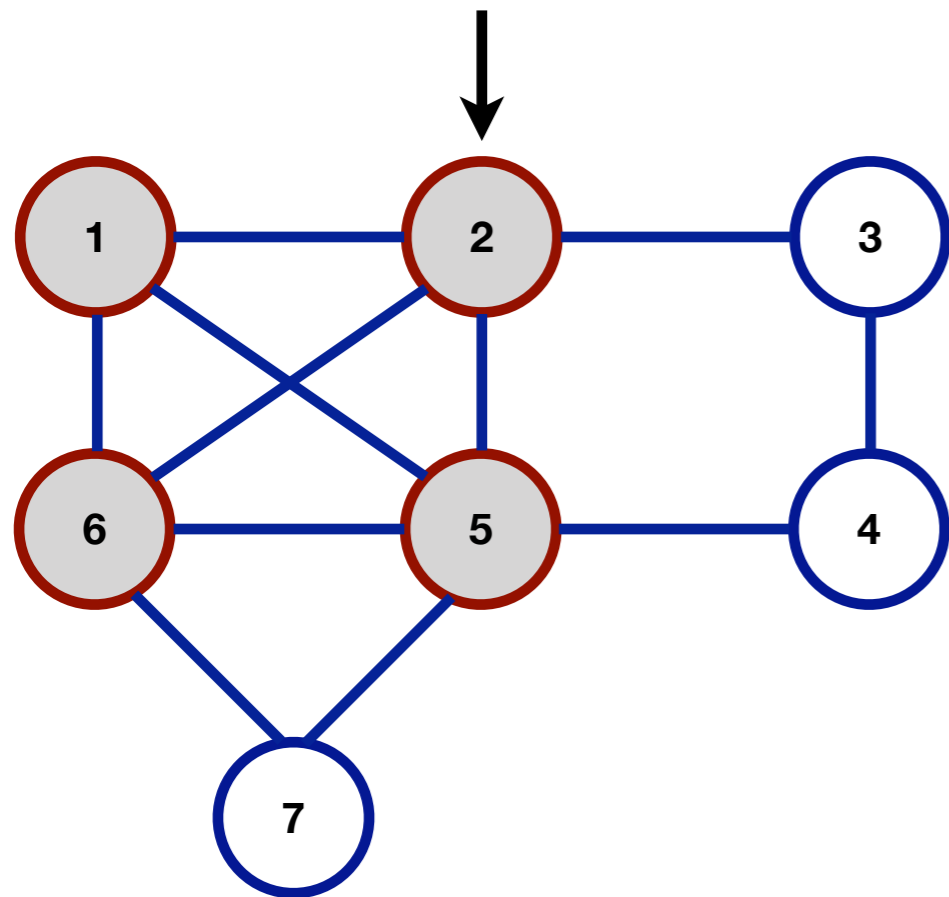


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

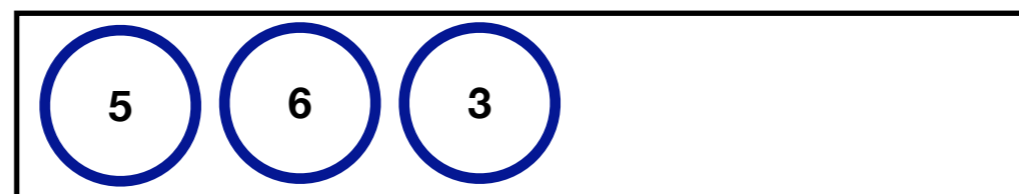
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2



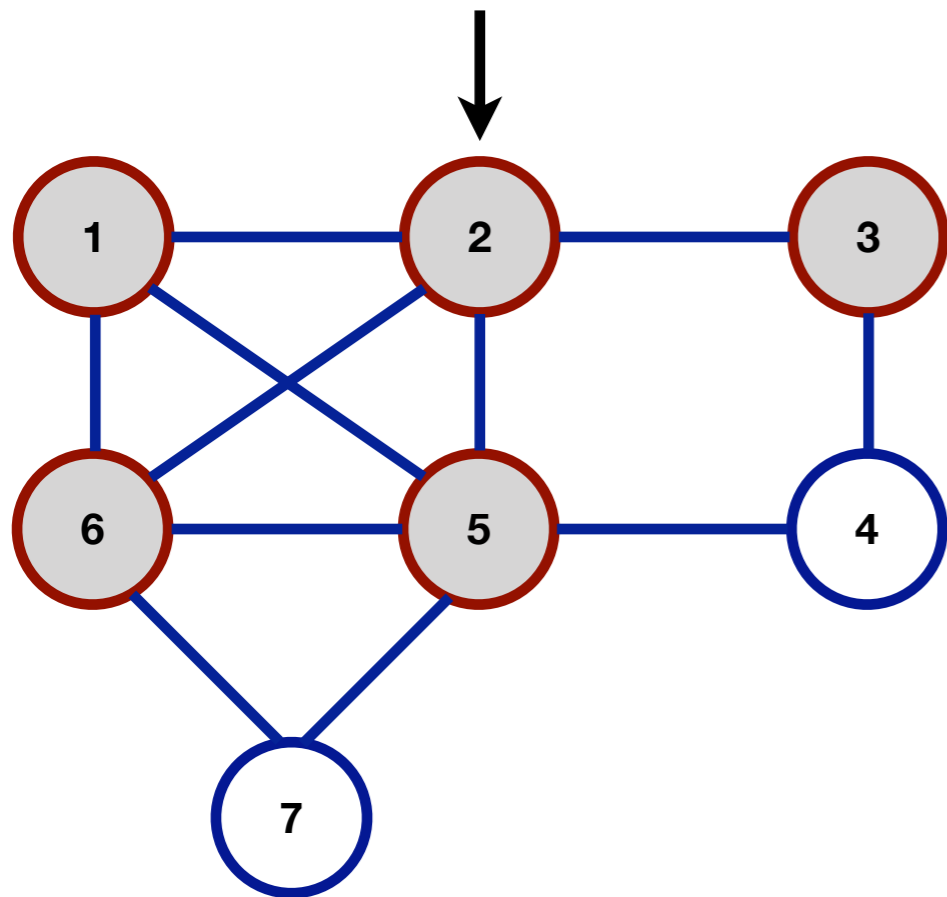
```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)

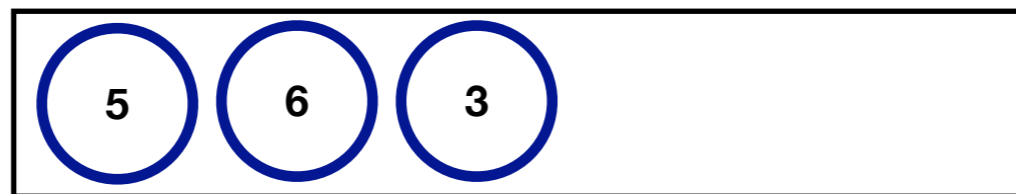


```

proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
  
```

**n.processed := true**

1 2



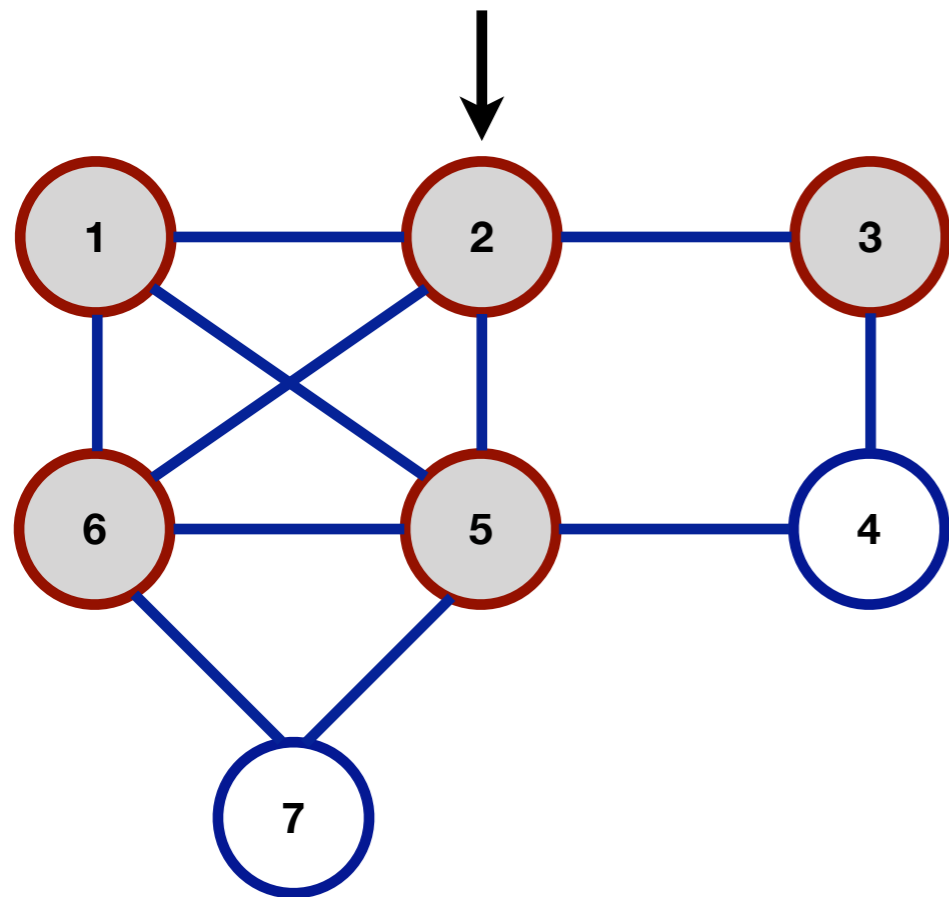
```

[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
  
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)

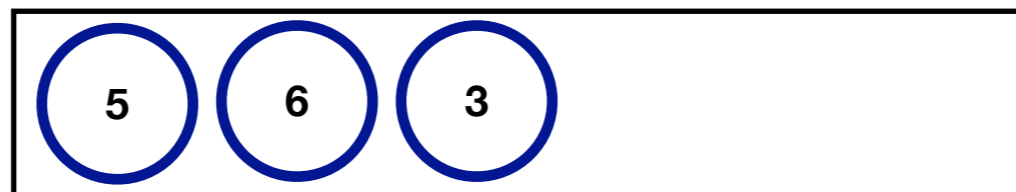


```

proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
  
```

**endif**

1 2

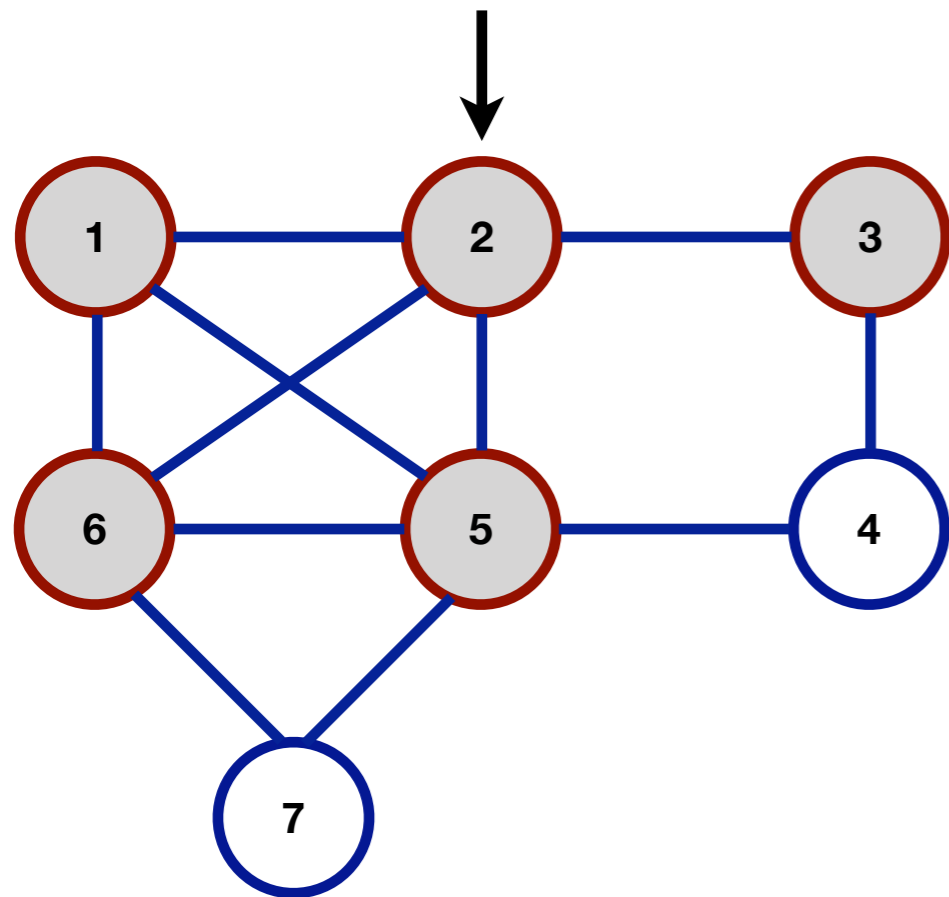


[1]	2	5	6
[2]	1	3	5 6
[3]	2	4	
[4]	3	5	
[5]	1	2	4 6 7
[6]	1	2	5 7
[7]	5	6	

Adjacency List

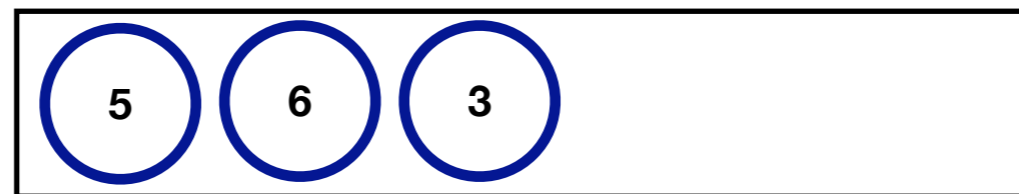
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2

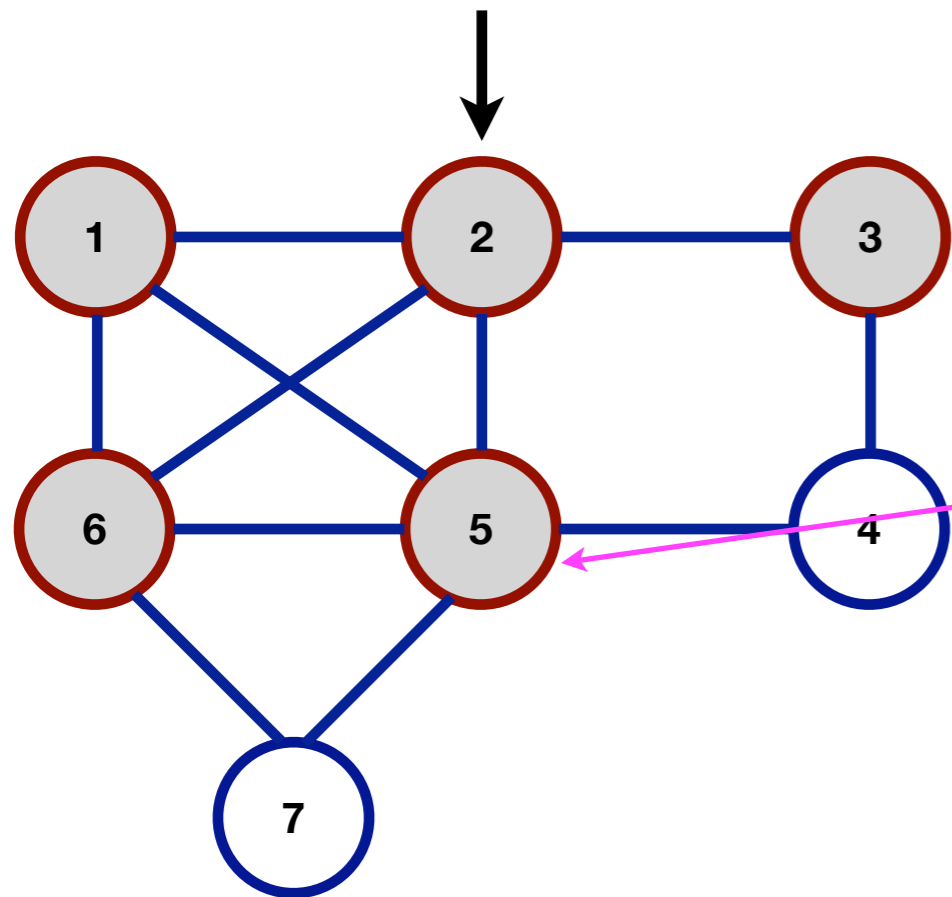


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

for n in cv.neighbors[]

```
if (not n.processed)
  Q.enqueue(n)
  n.processed := true
```

```
endif
```

```
endfor
```

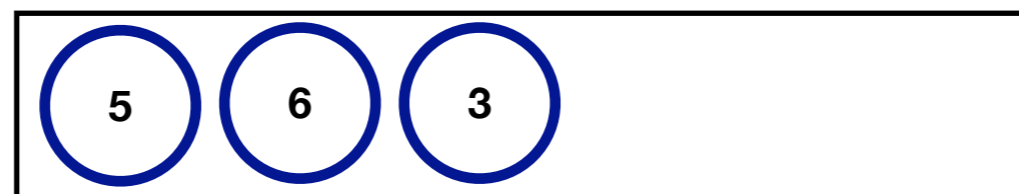
```
endwhile
```

```
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

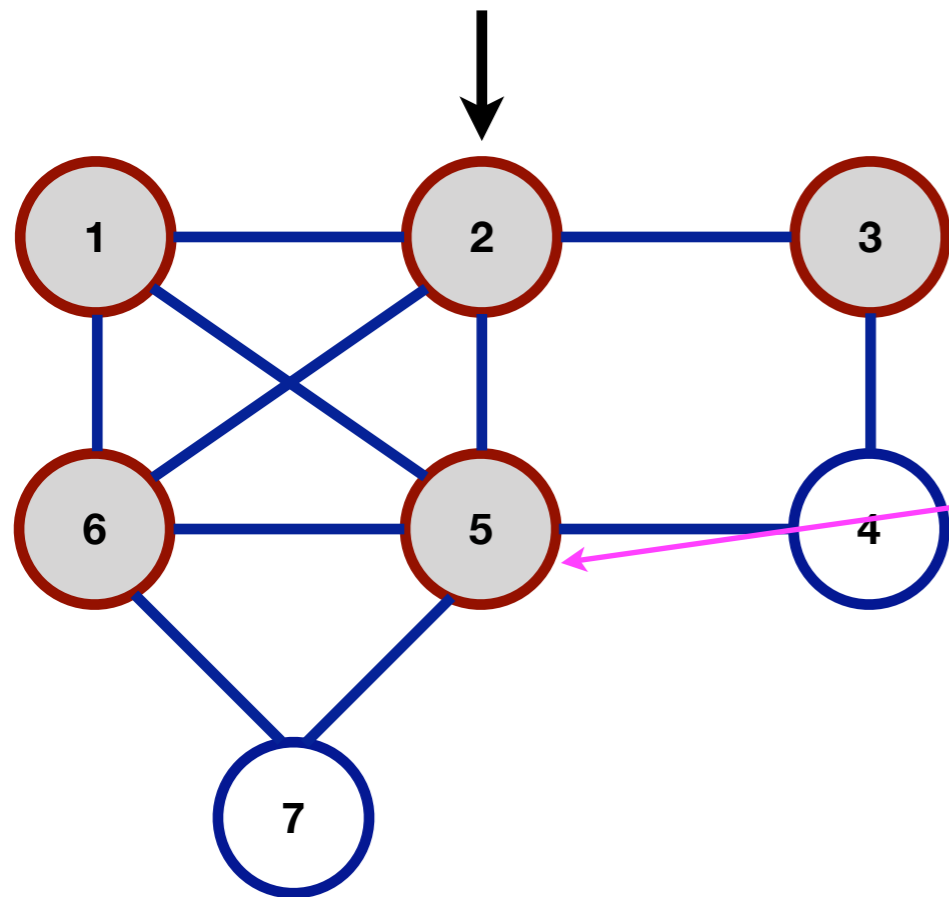
Adjacency List

1 2



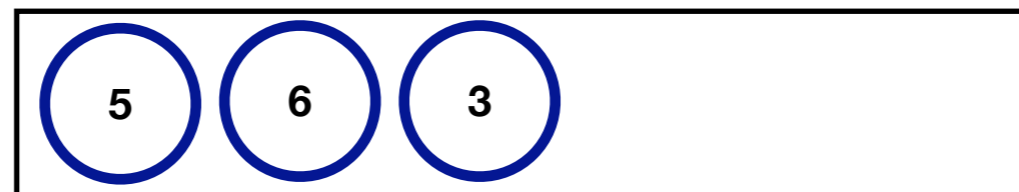
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2

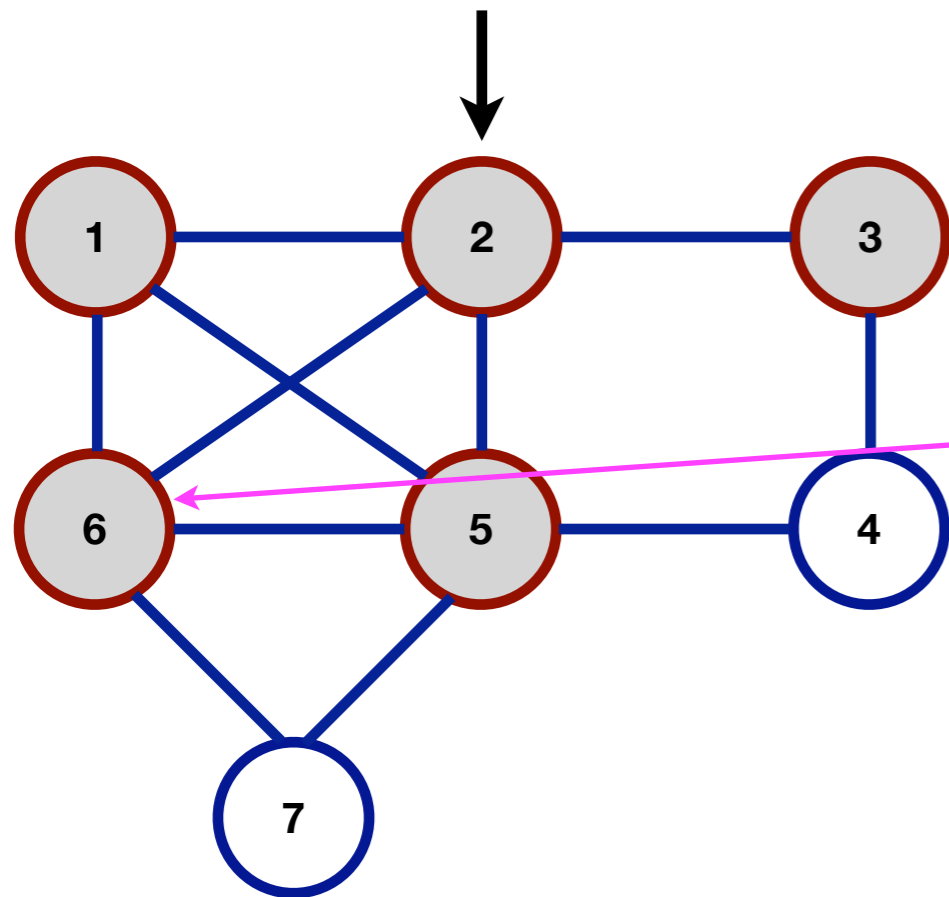


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

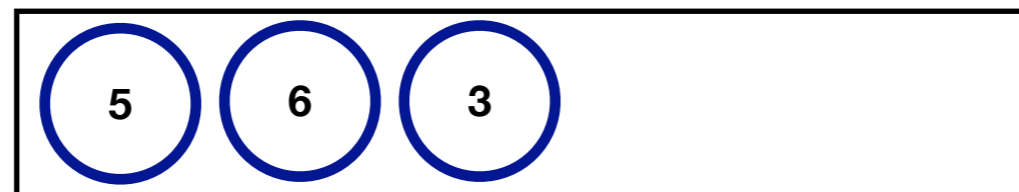
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2

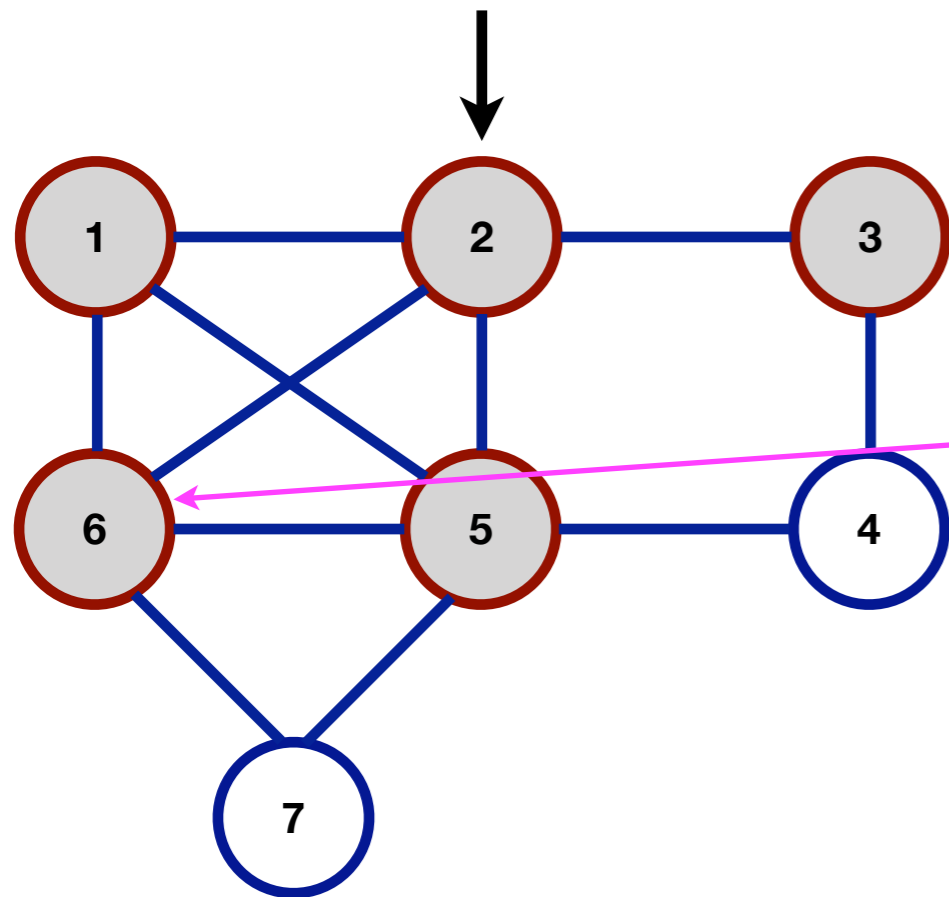


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endfor

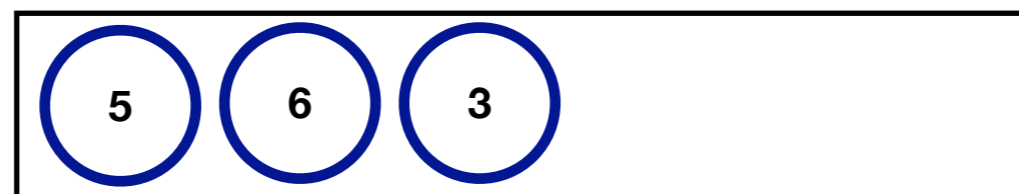
endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

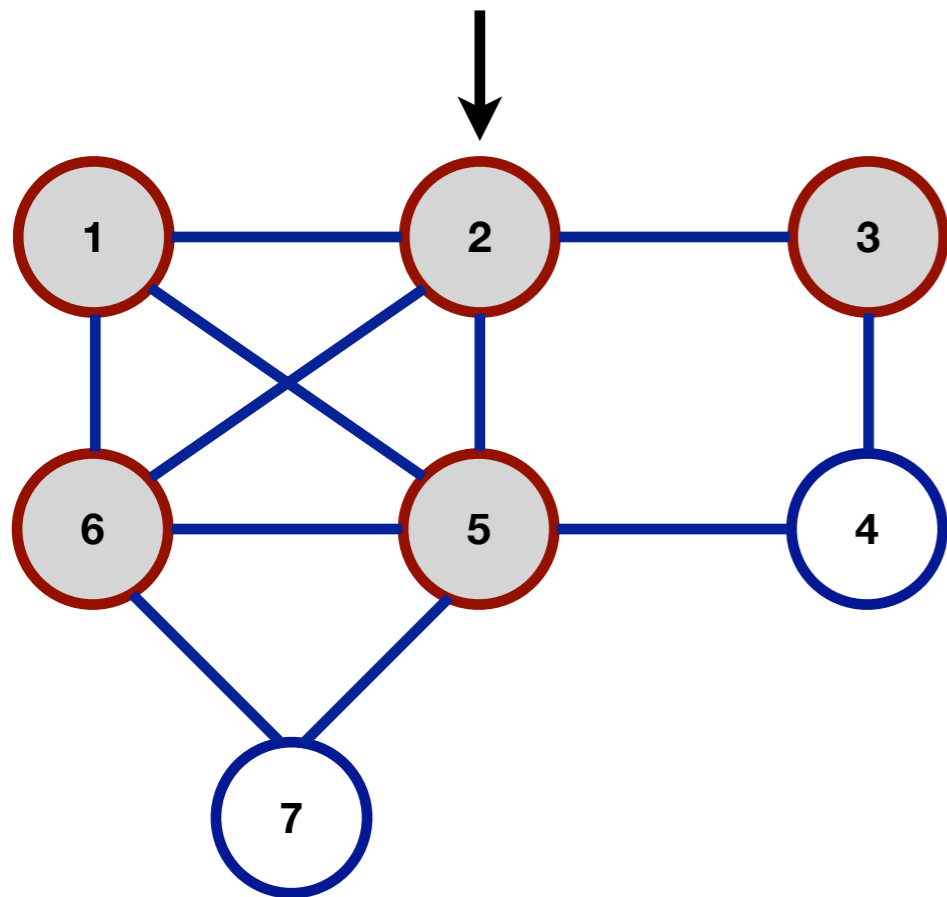
Adjacency List

1 2



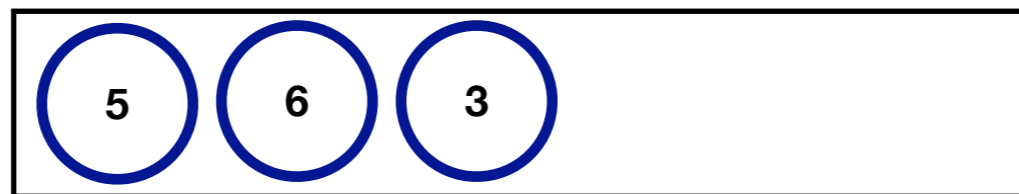
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

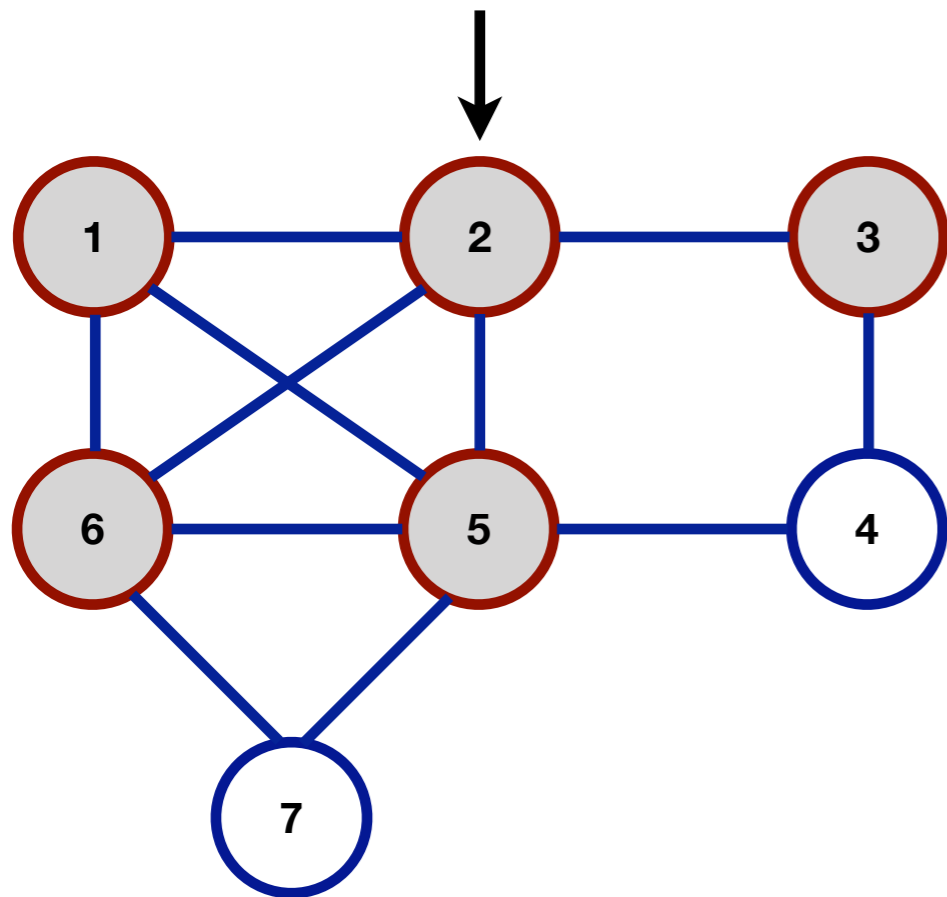
1 2





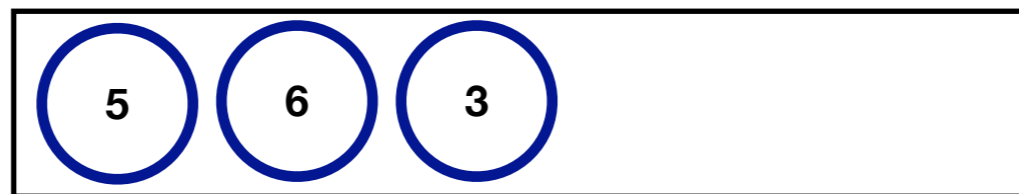
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



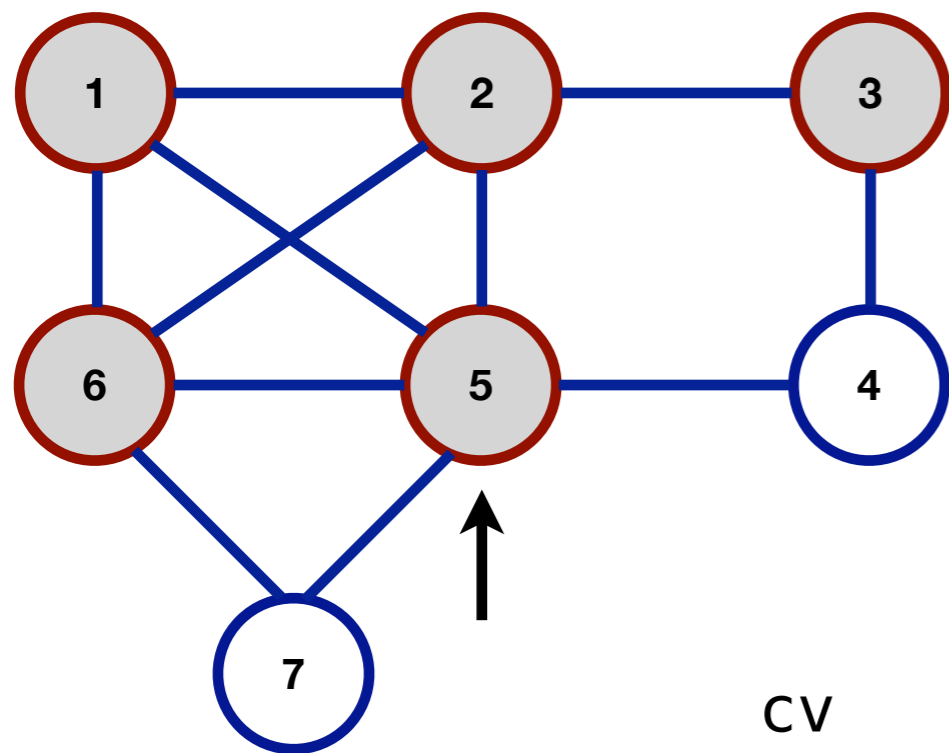
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2

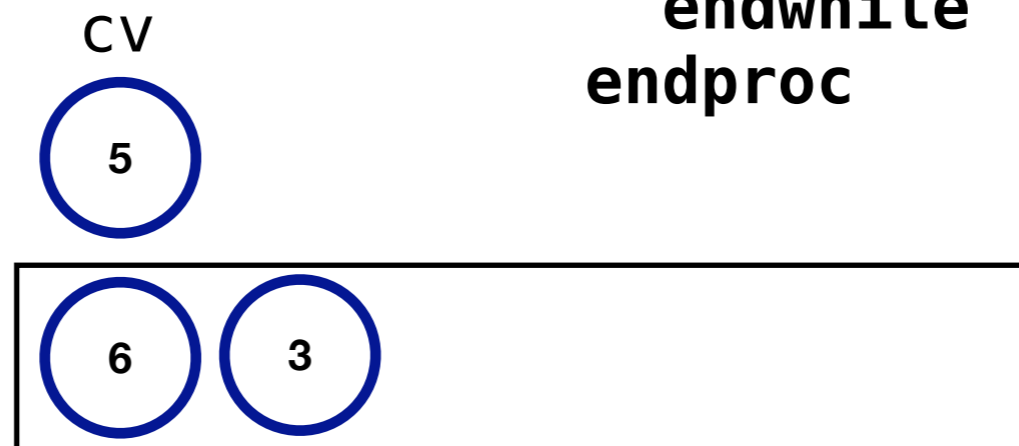


# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



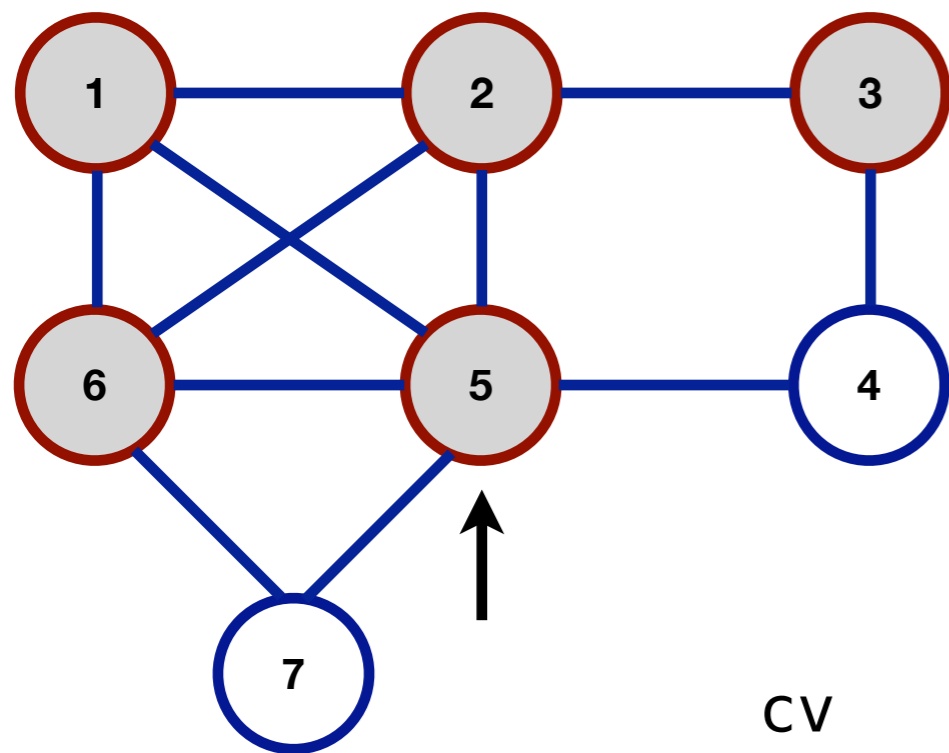
1 2



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



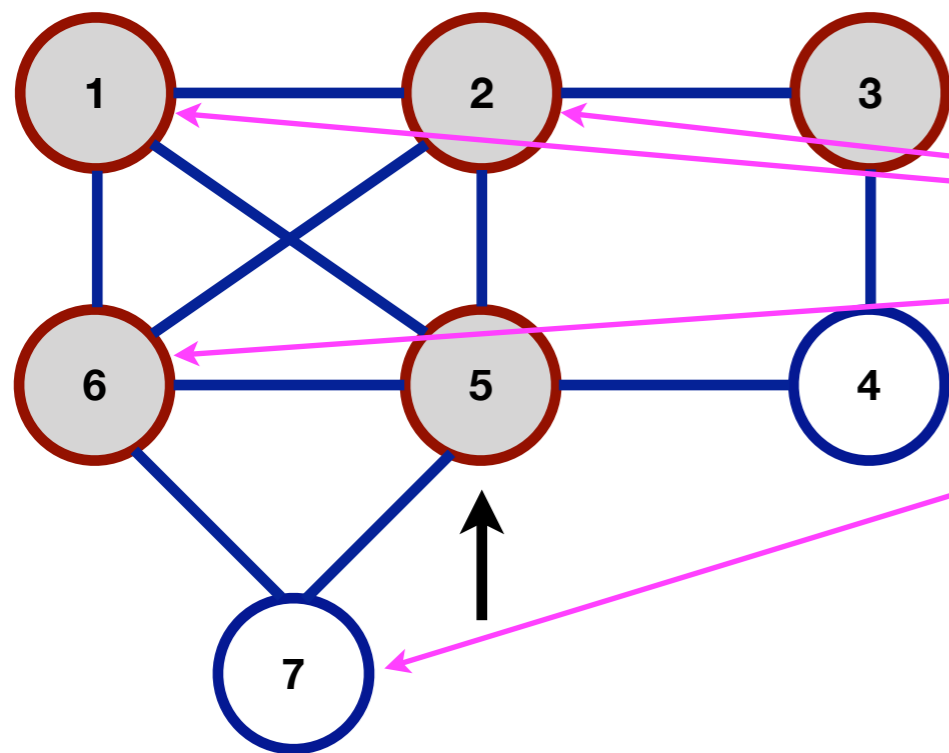
1 2 5



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

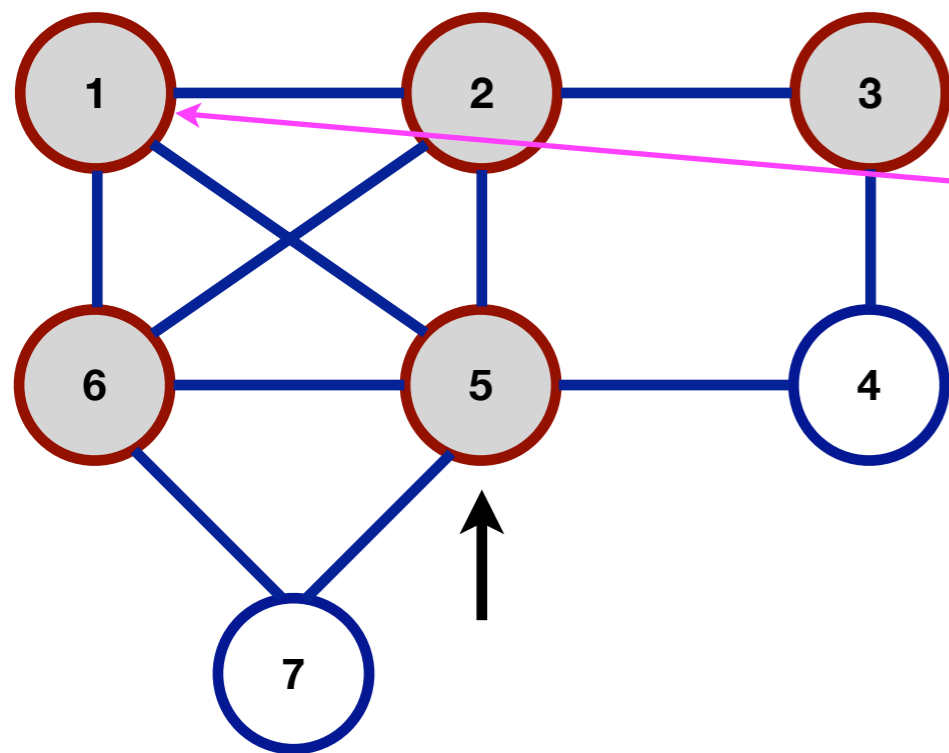


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

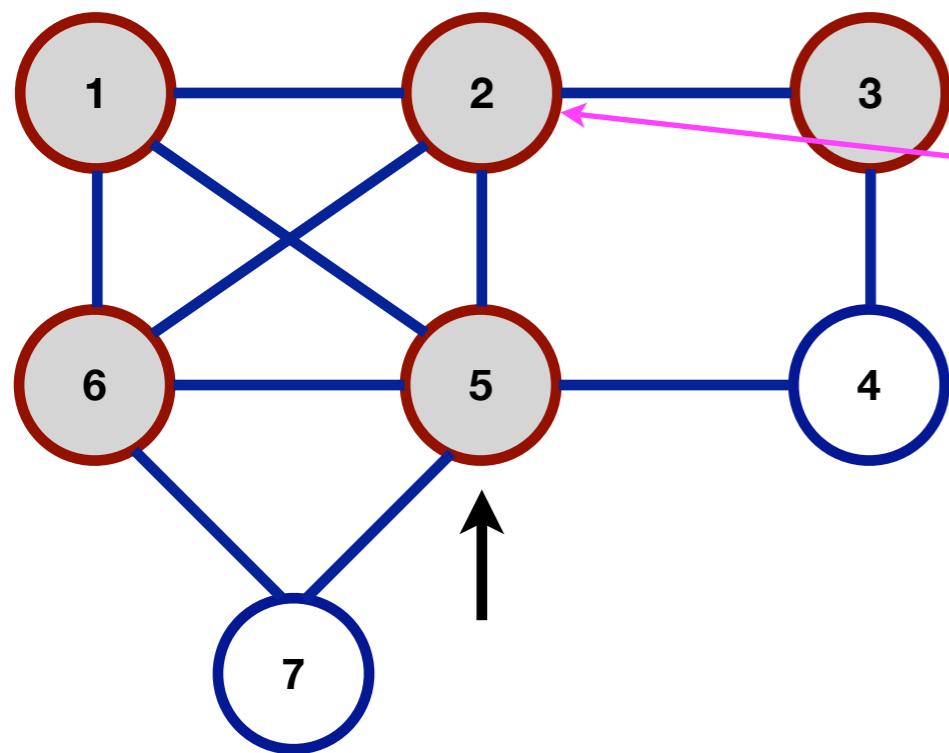


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

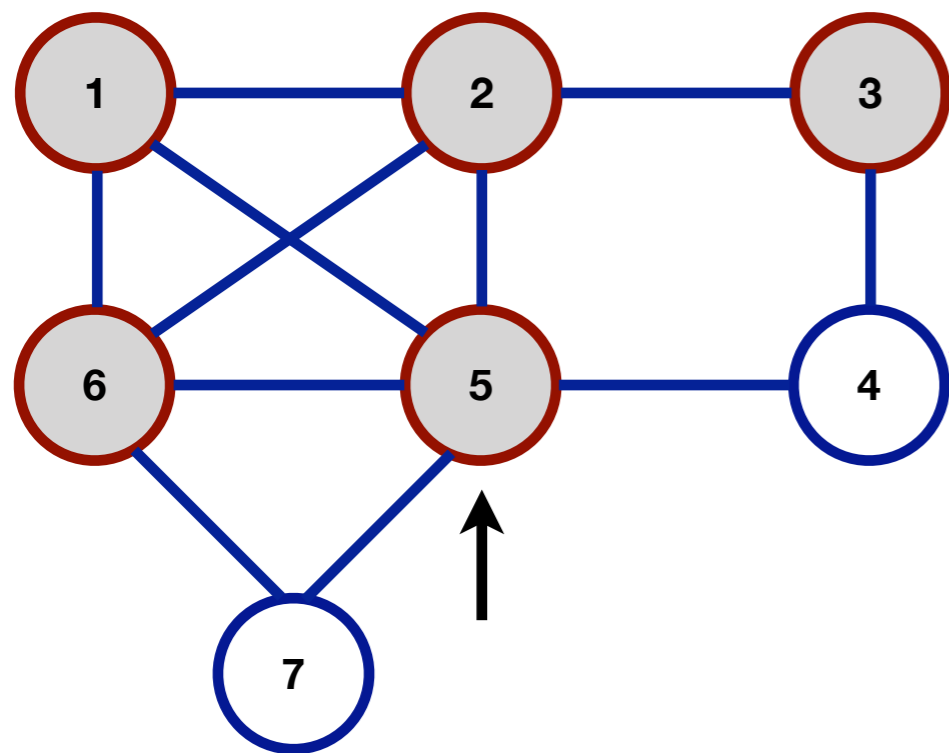


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

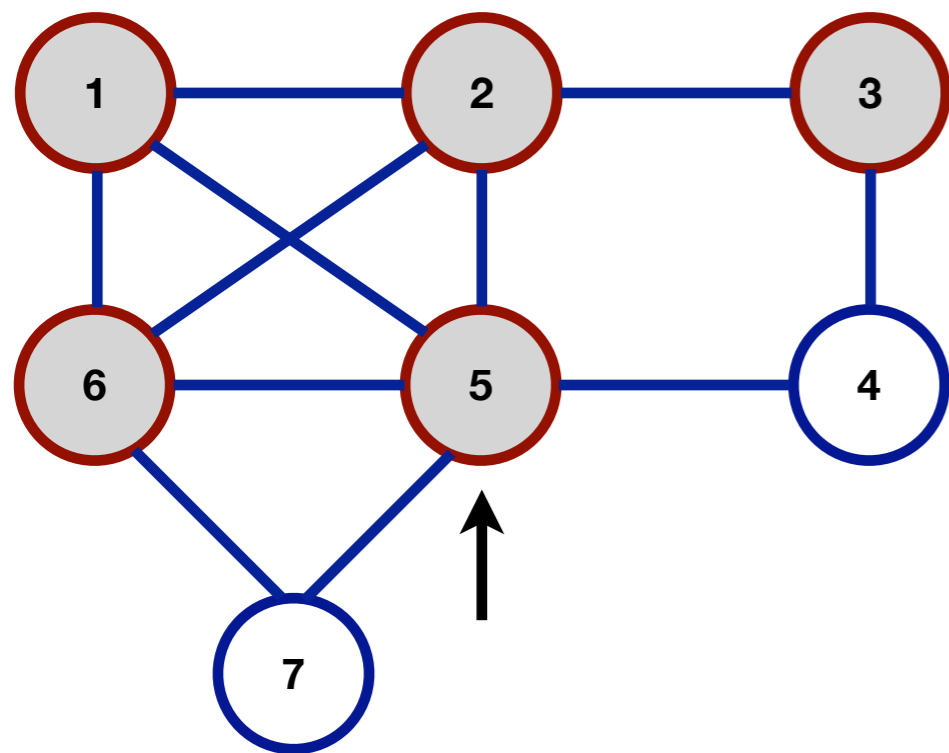


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

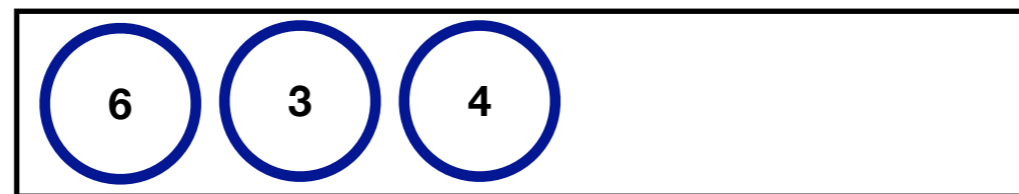
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5



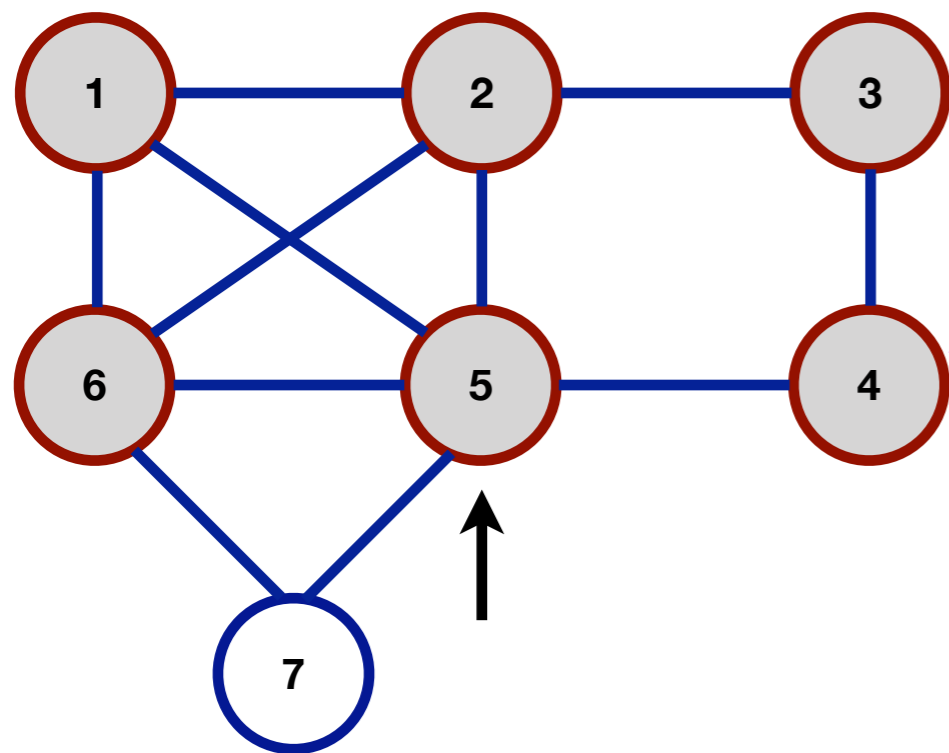
```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List



# Breadth-First Search / Traversal

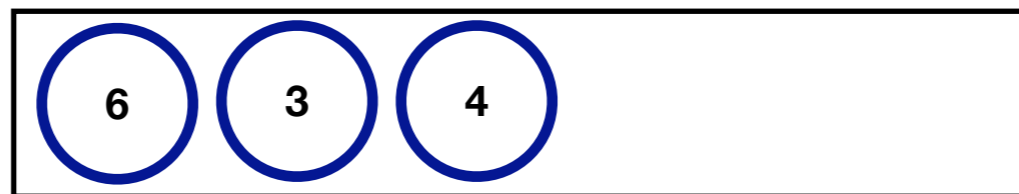
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

n.processed := true

1 2 5

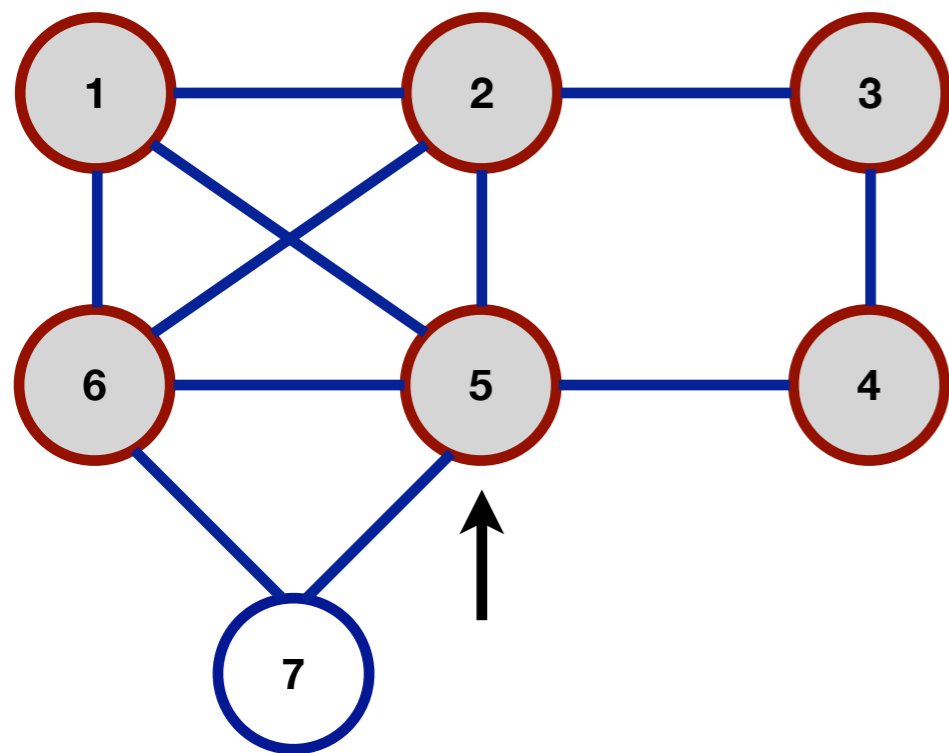


[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)

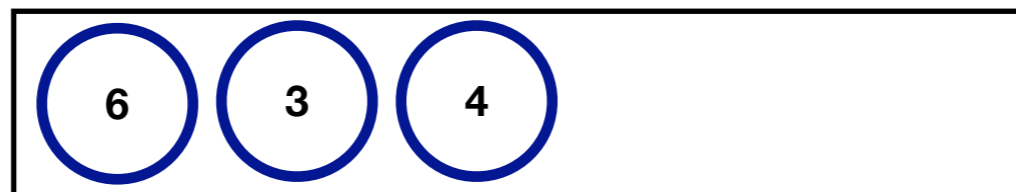


```

proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
  
```

**endif**

1 2 5

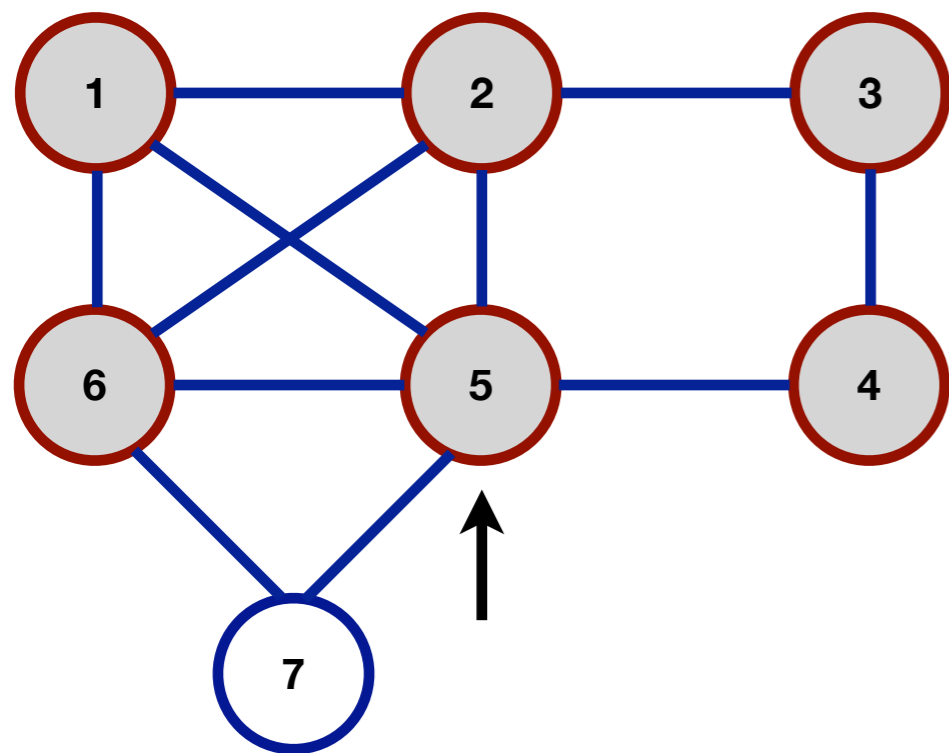


[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

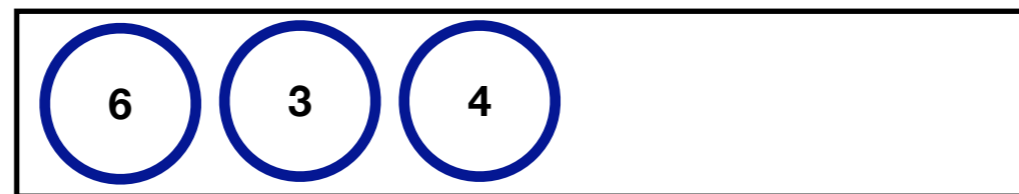
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

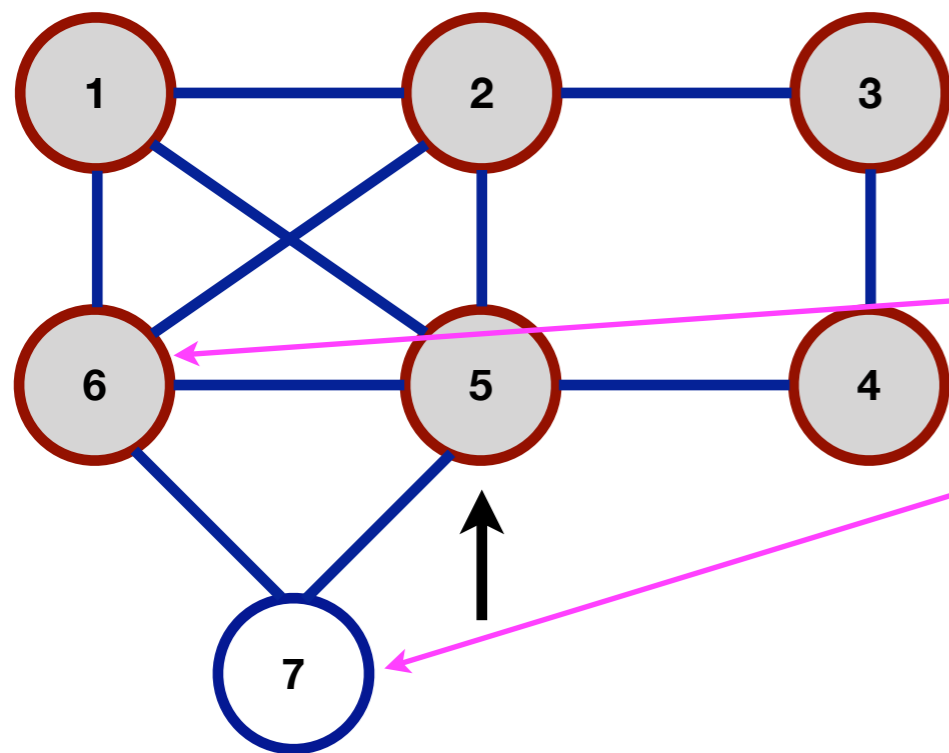


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

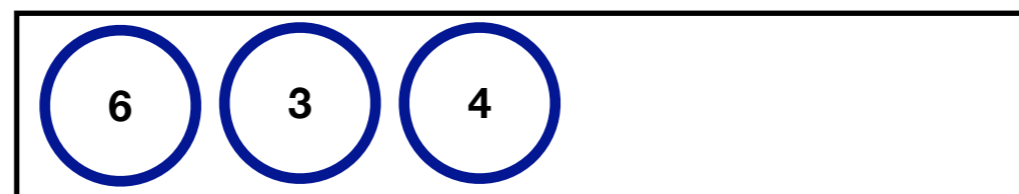
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

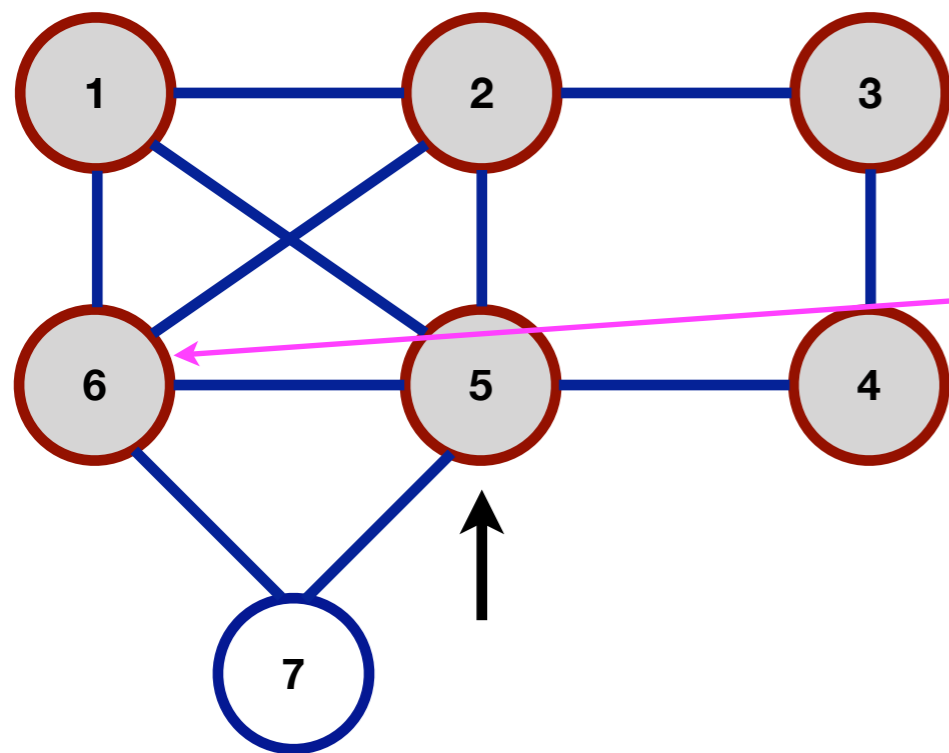


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

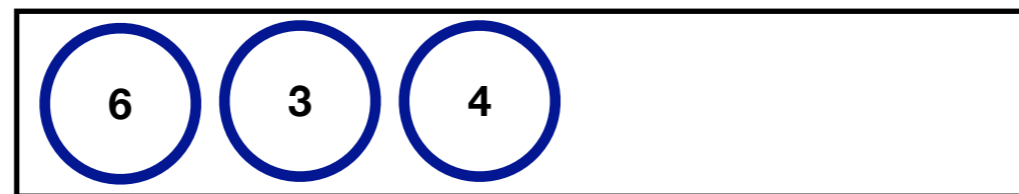
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

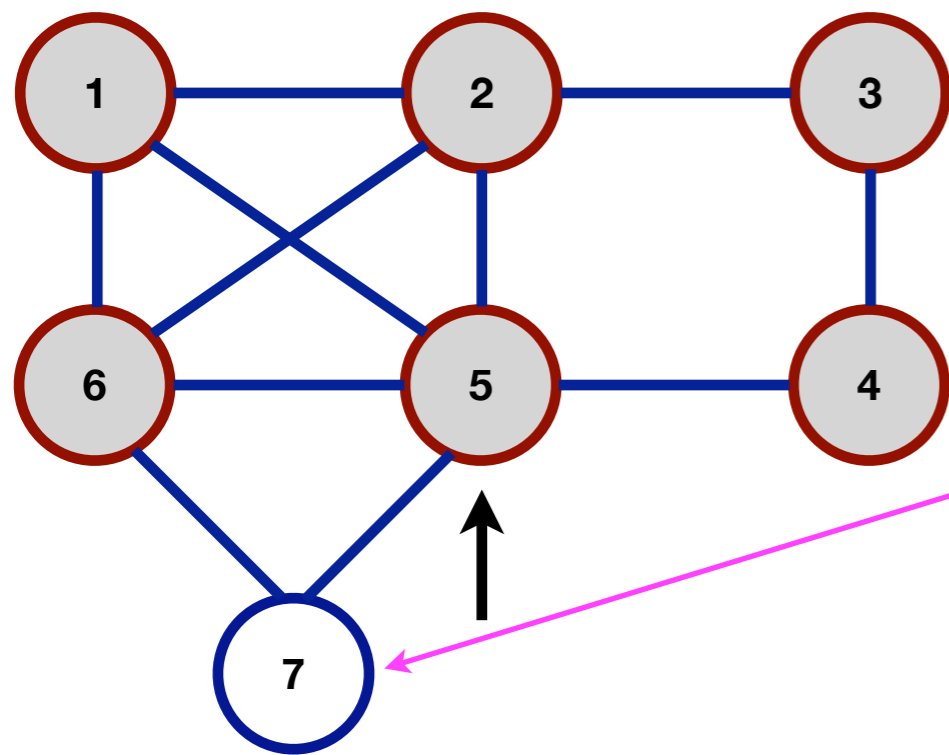


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

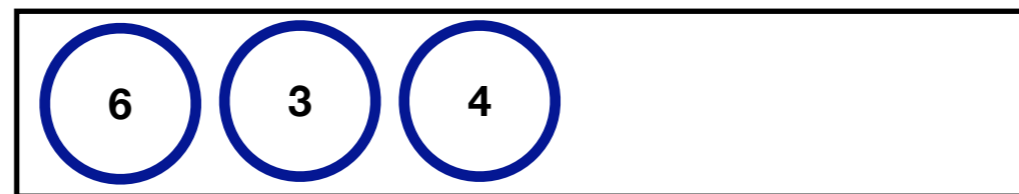
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

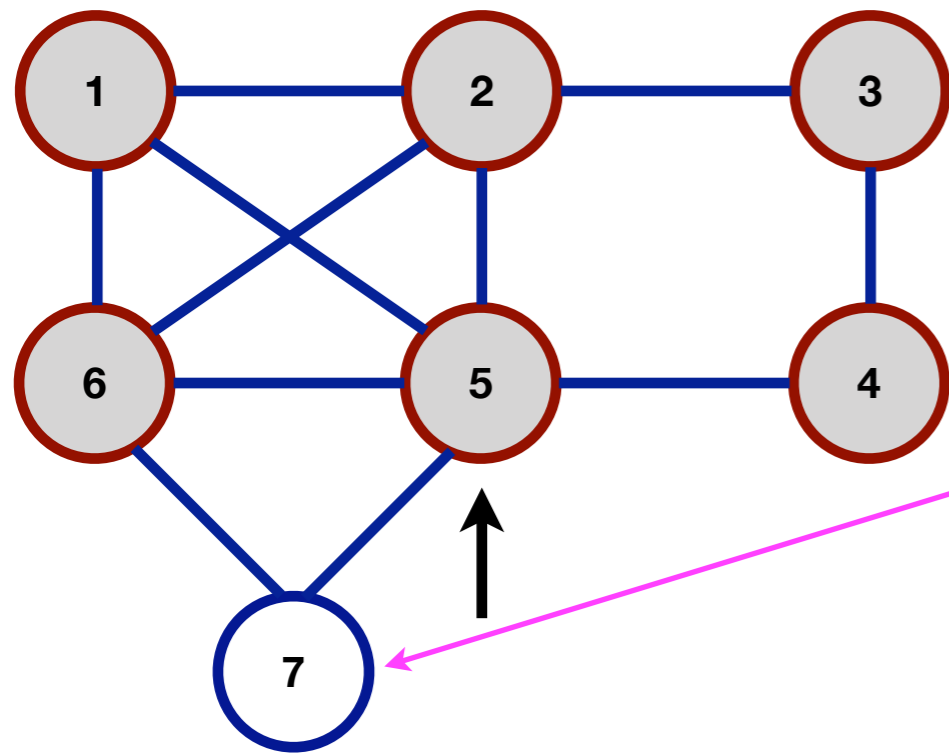


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

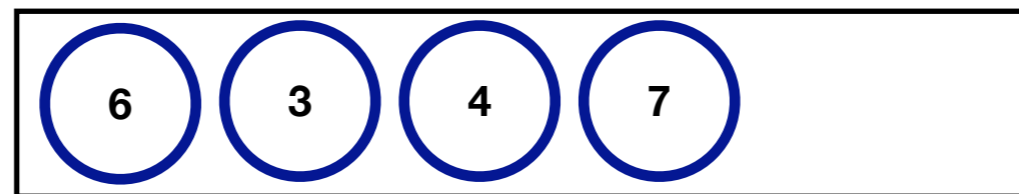
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

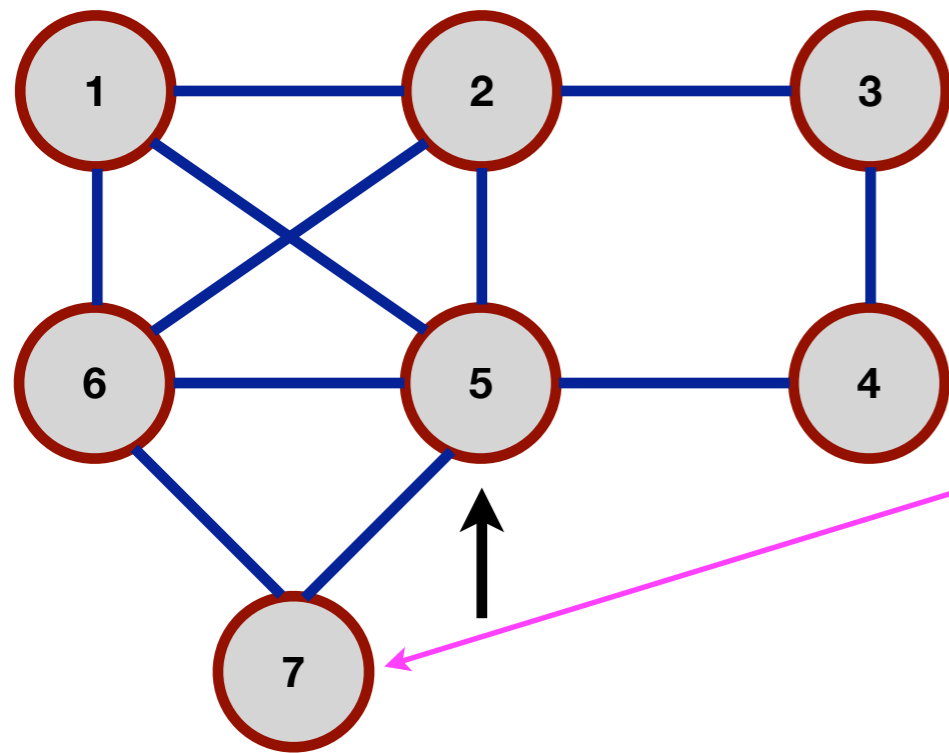


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

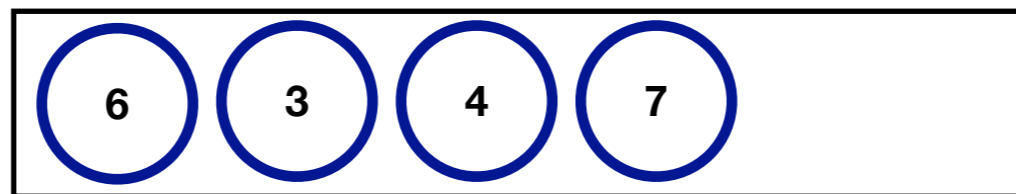
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

n.processed := true

1 2 5



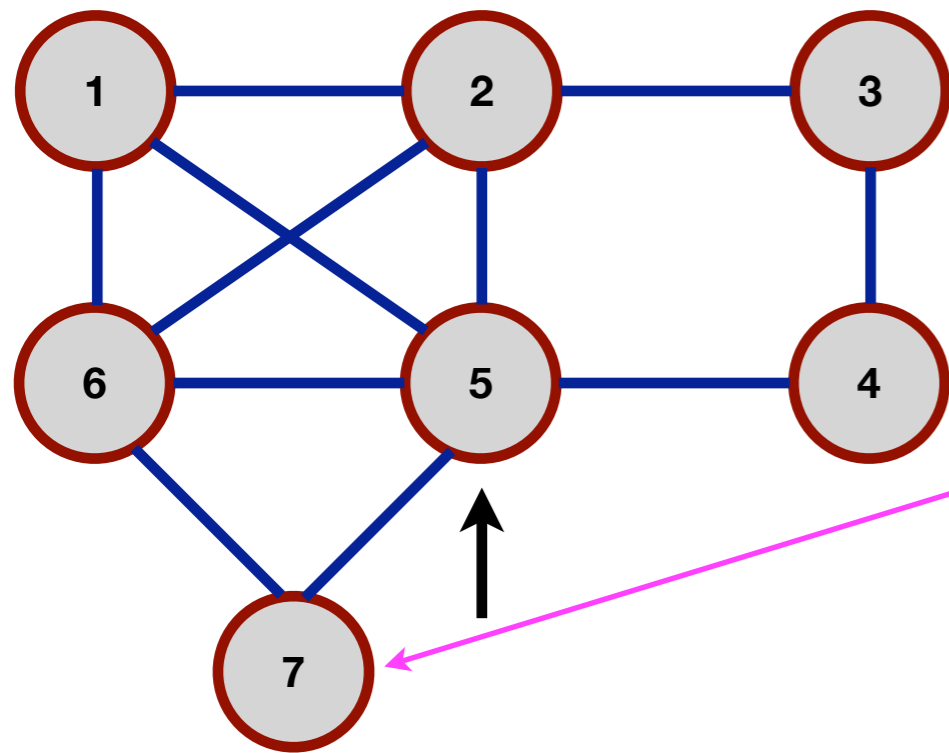
[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endif

endfor

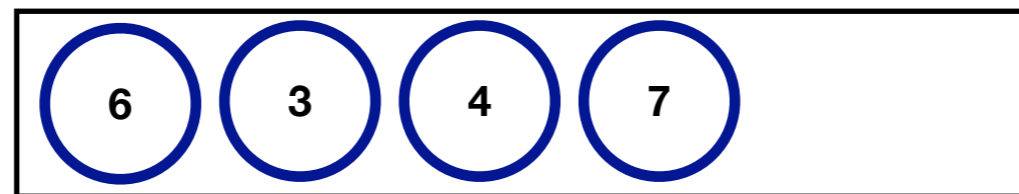
endwhile

endproc

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

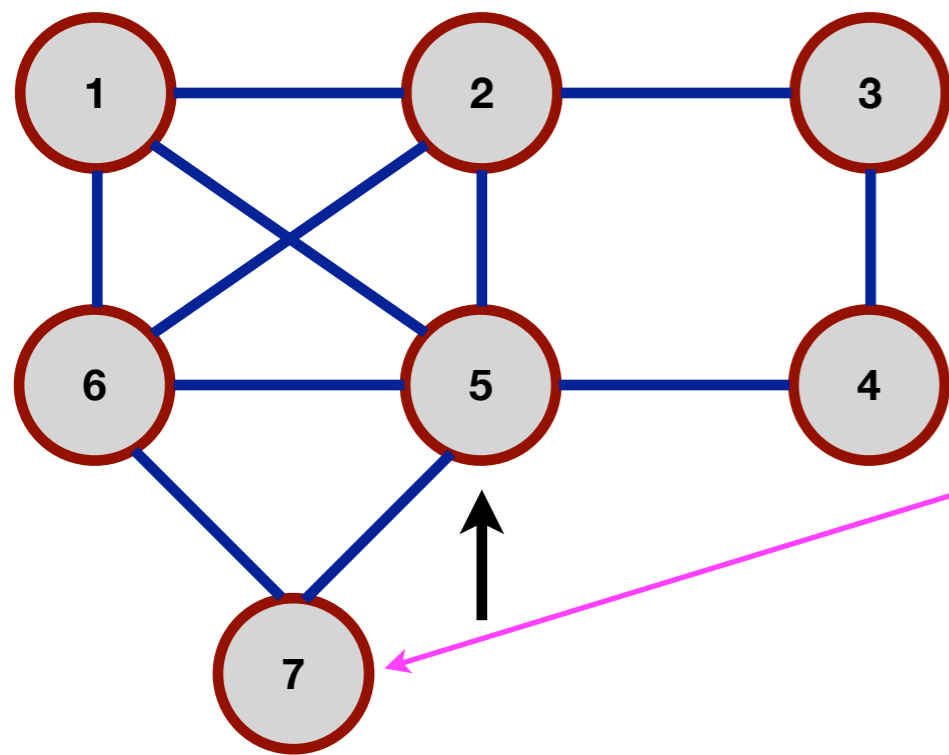
Adjacency List

1 2 5



# Breadth-First Search / Traversal

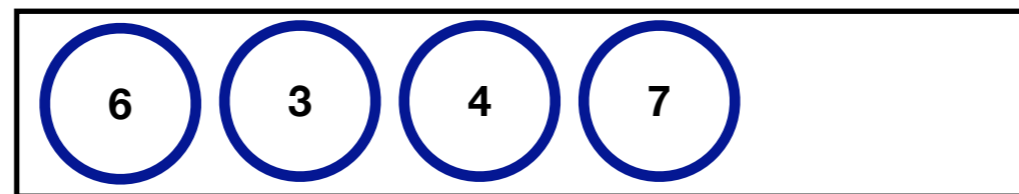
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

endfor

1 2 5

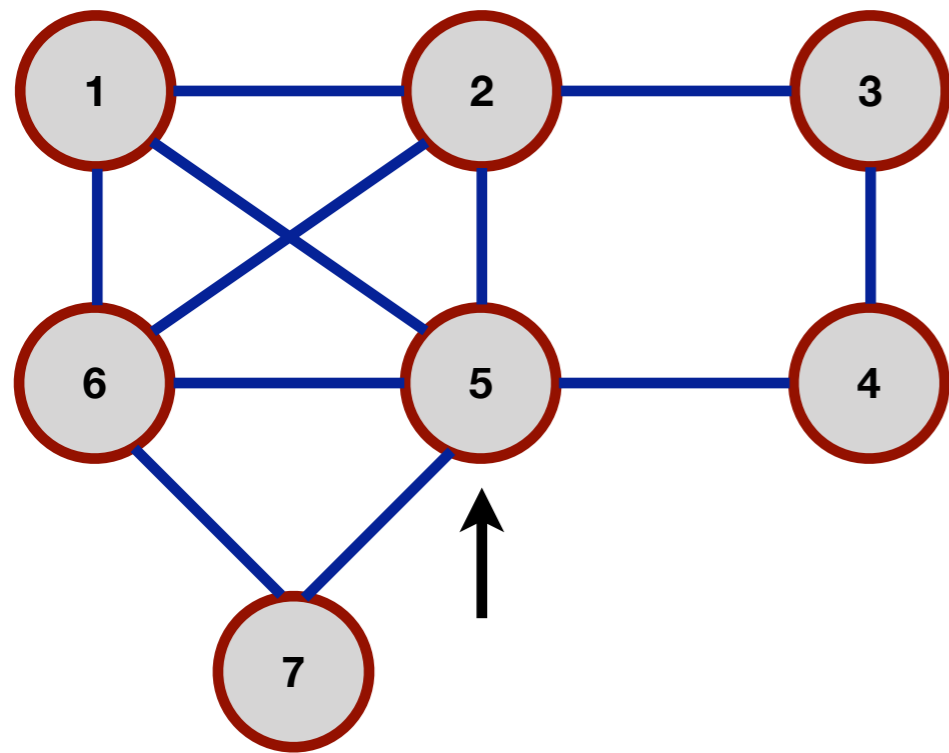


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

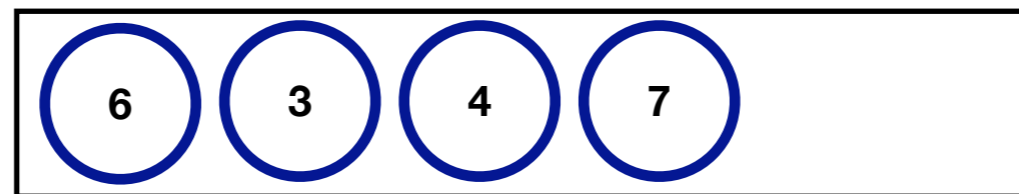
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



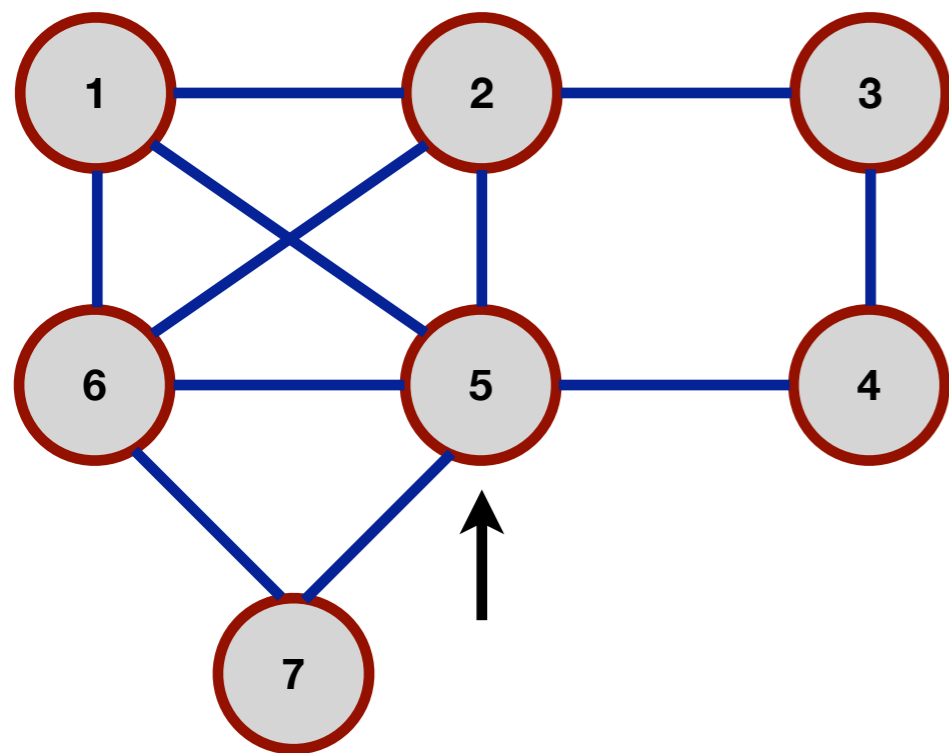
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5



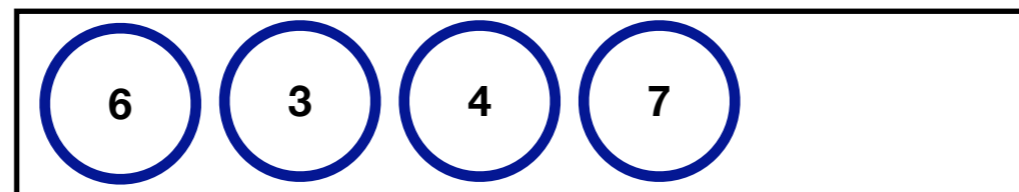
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



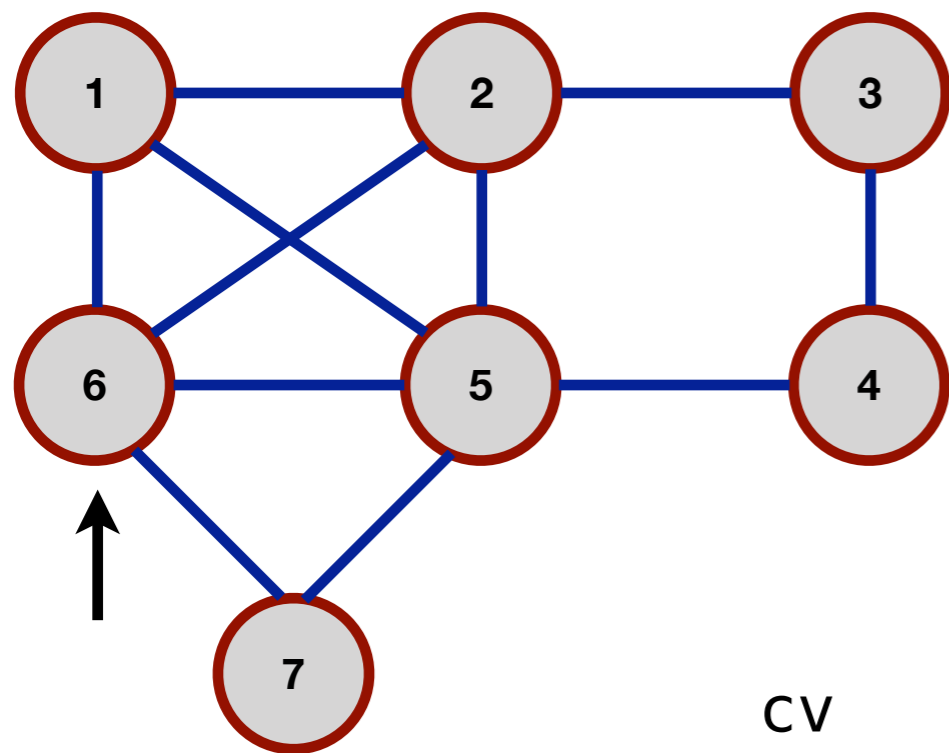
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5

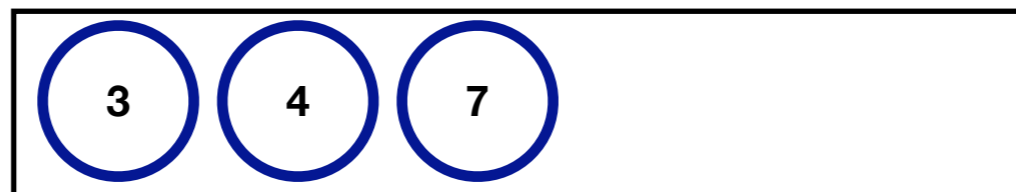
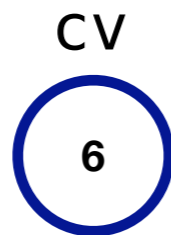


# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



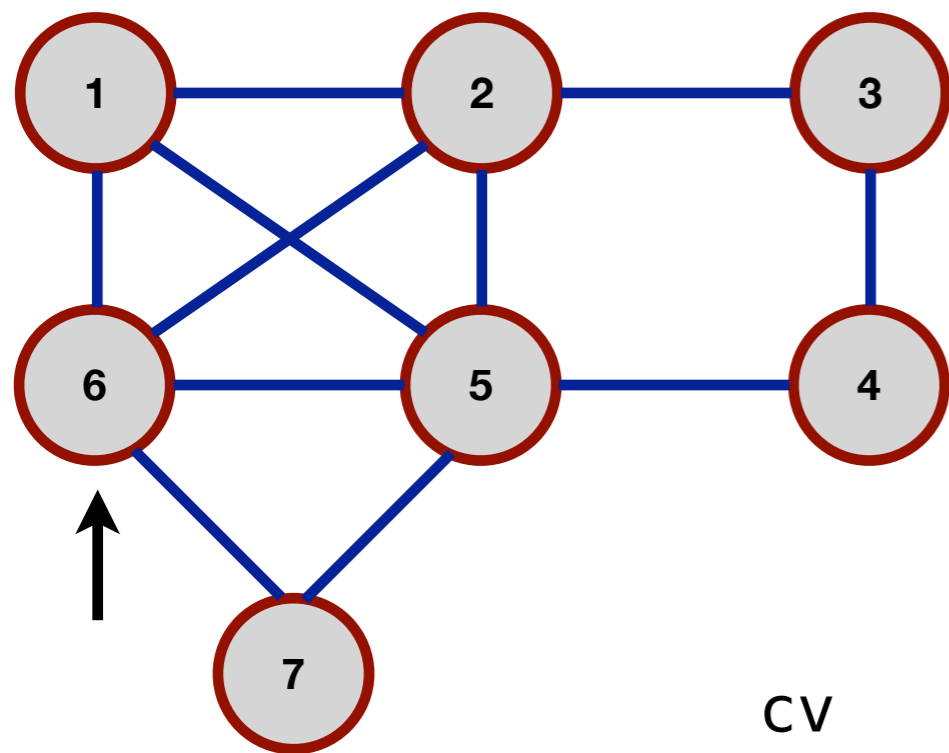
1 2 5



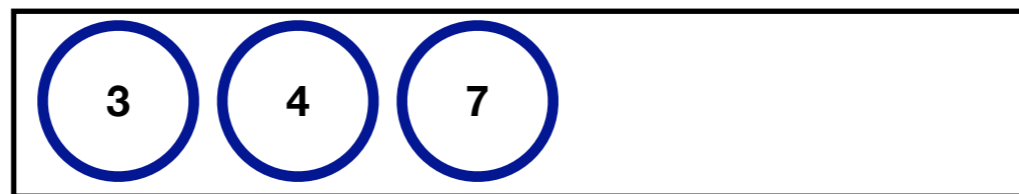
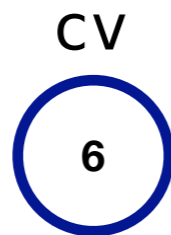
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



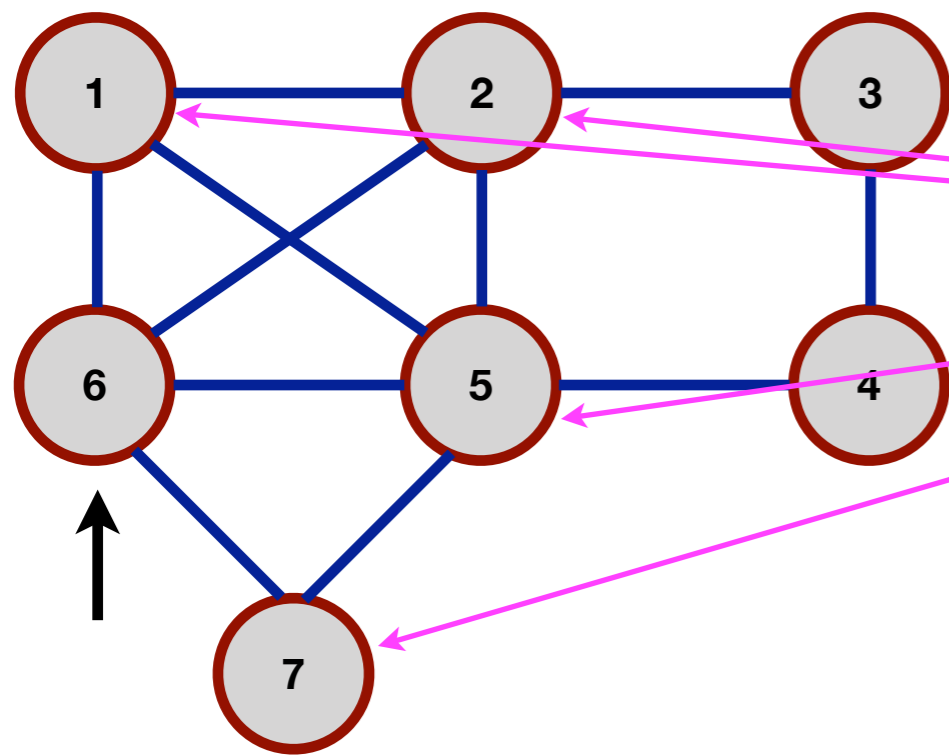
1 2 5 6



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

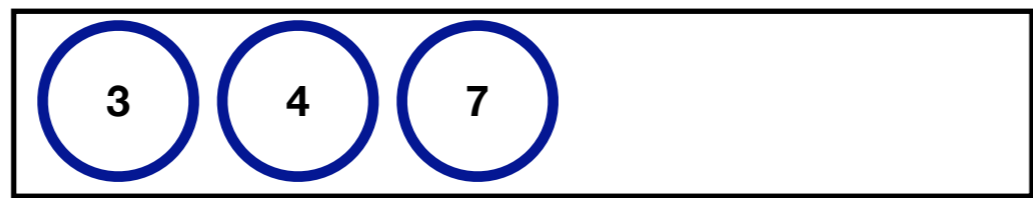
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

for n in cv.neighbors[]

1 2 5 6

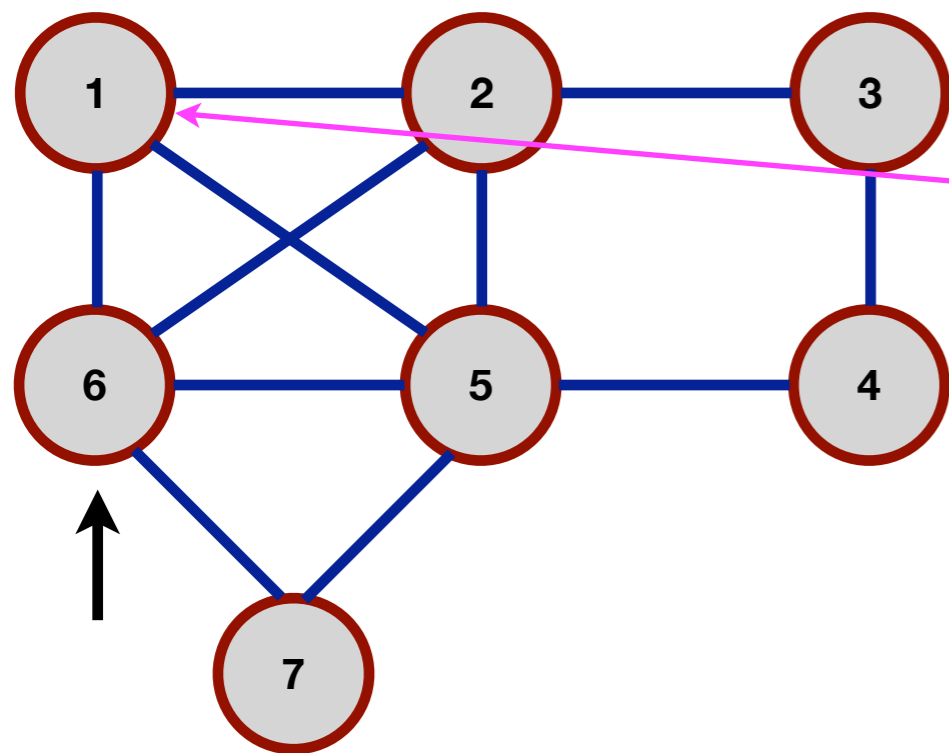


[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

# Breadth-First Search / Traversal

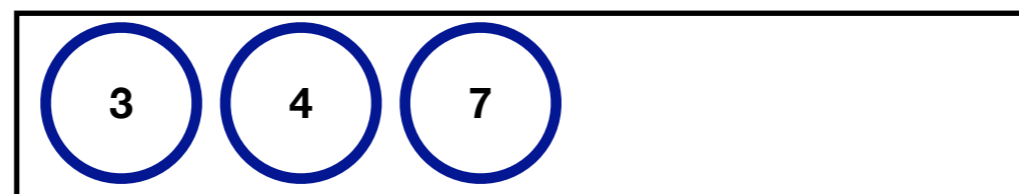
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1 2 5 6



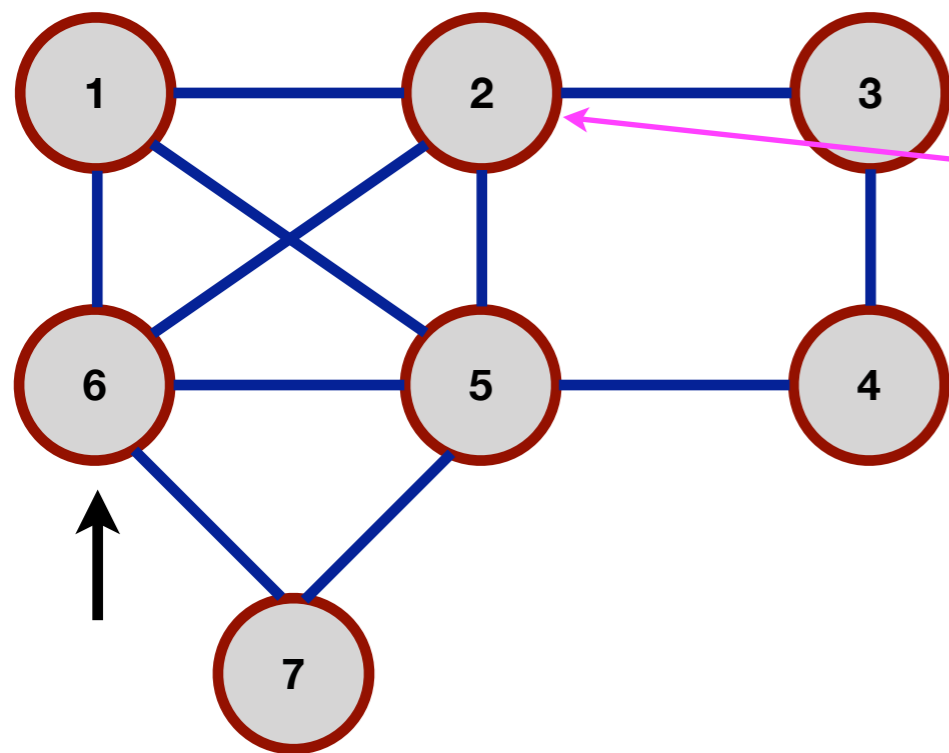
```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List



# Breadth-First Search / Traversal

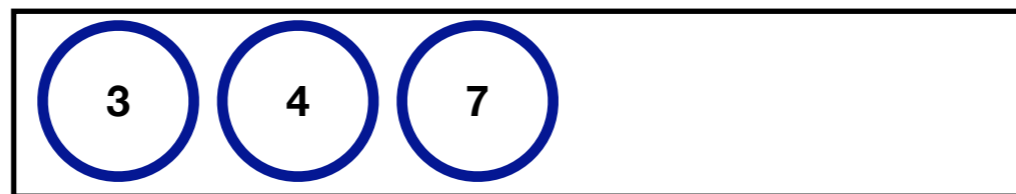
myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1 2 5 6

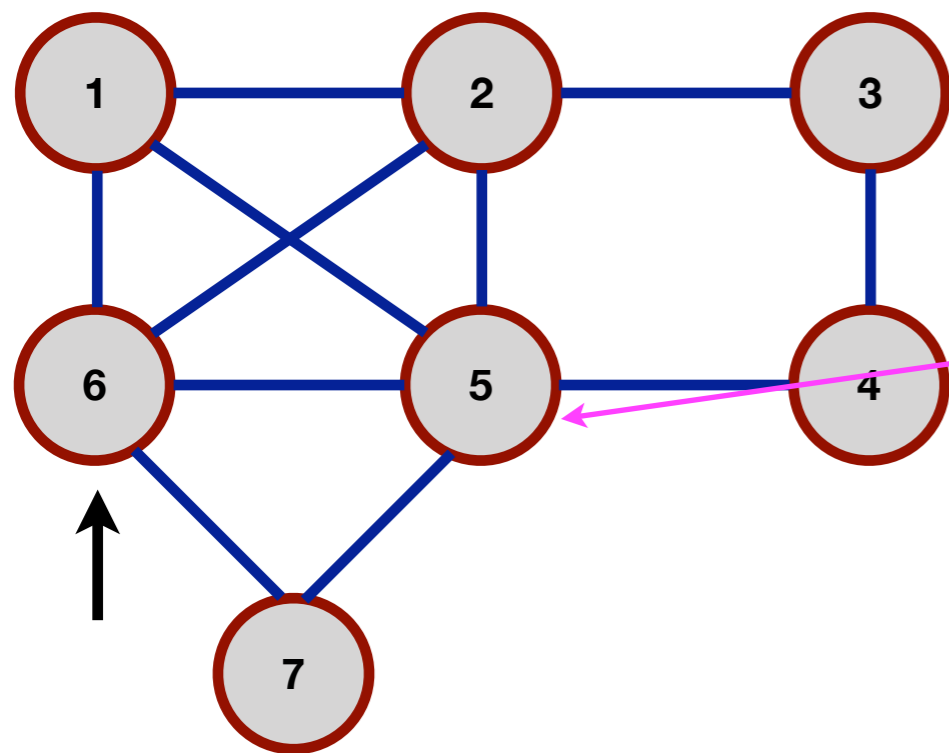


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

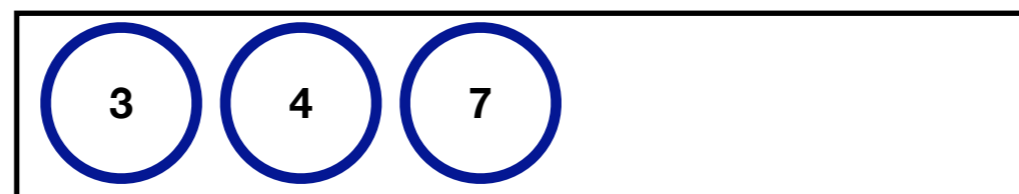
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6

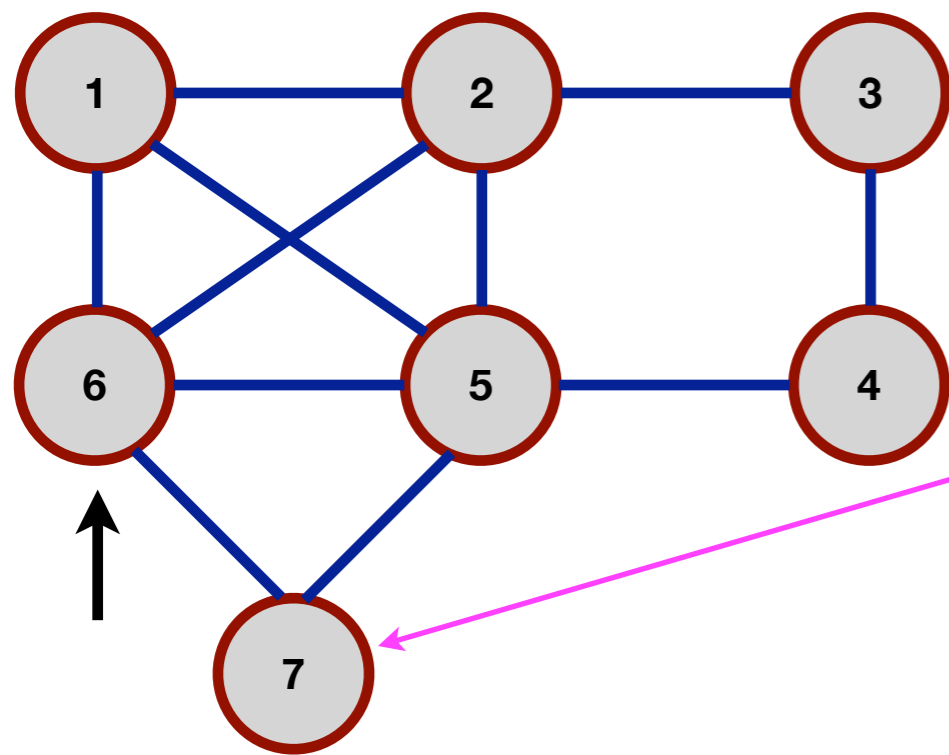


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

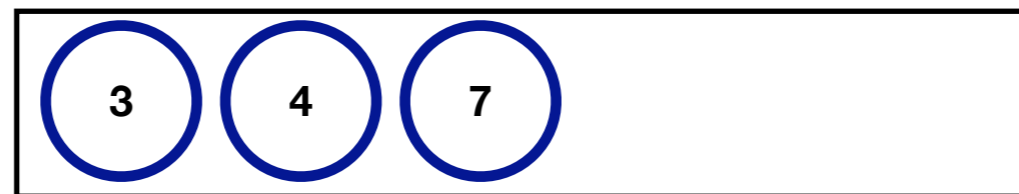
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6

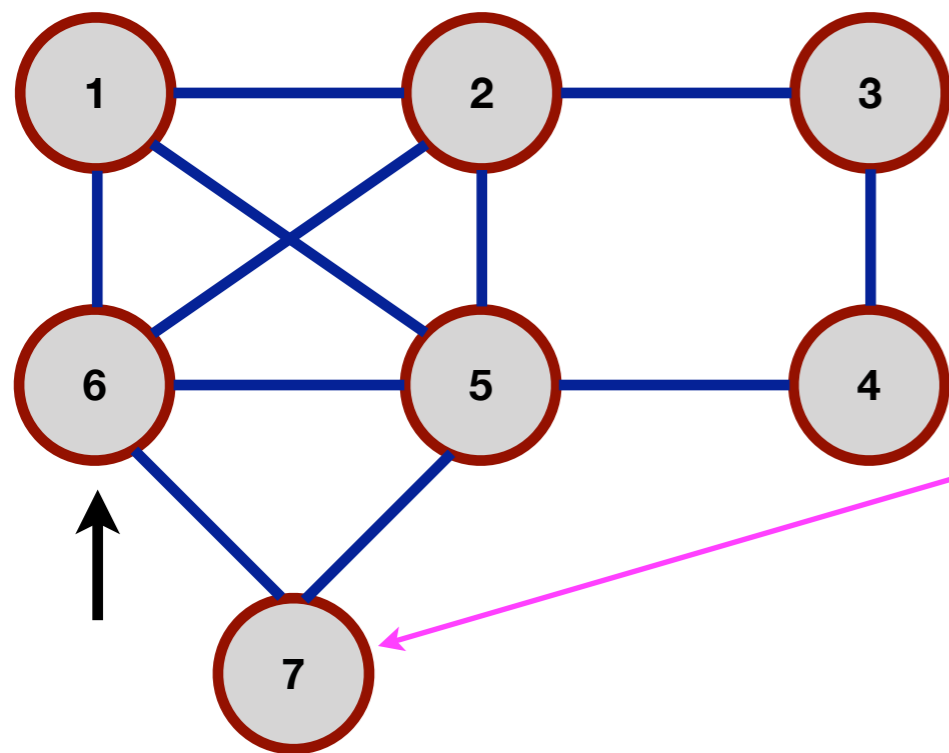


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

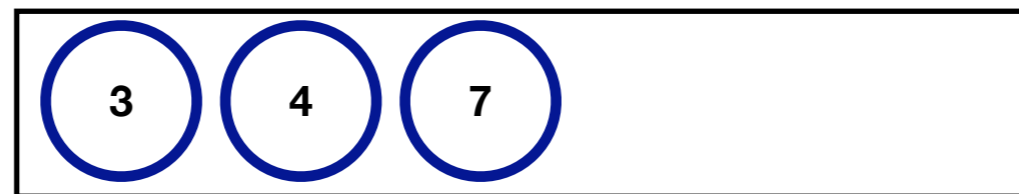
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6

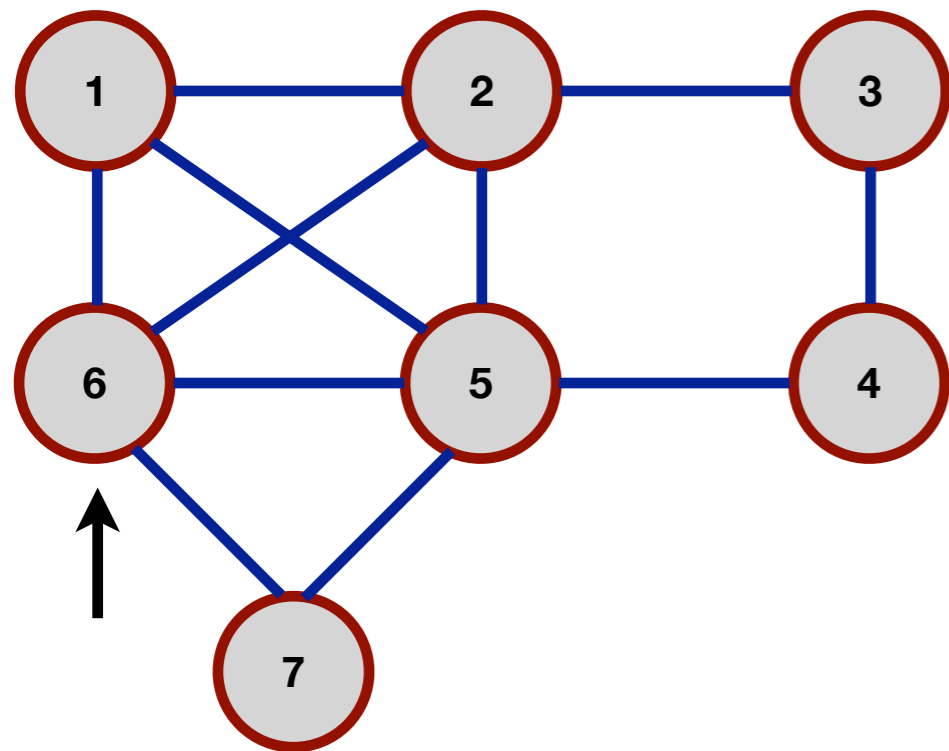


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

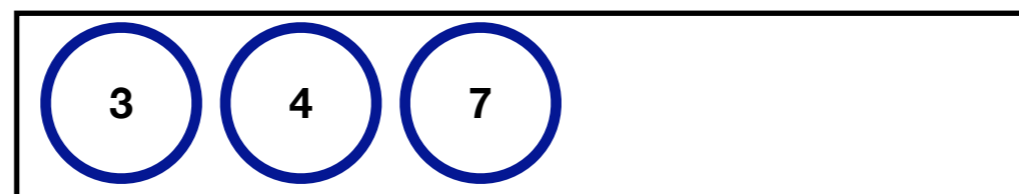
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



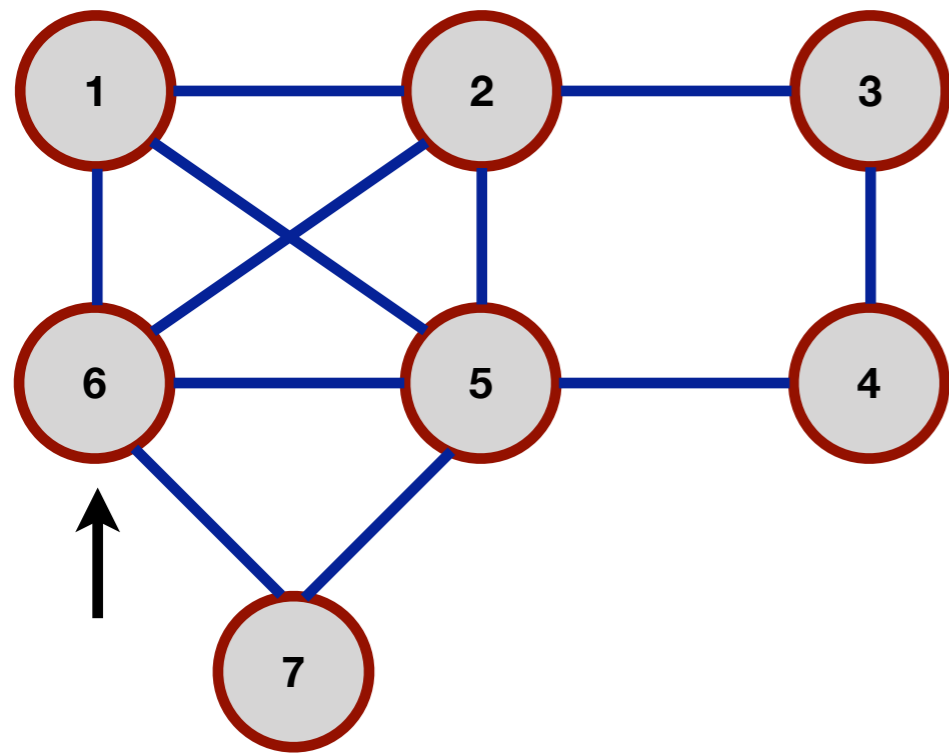
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6



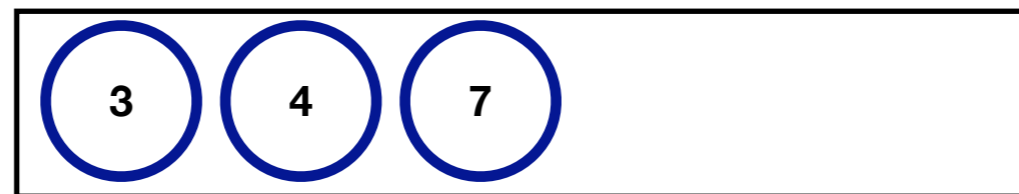
# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



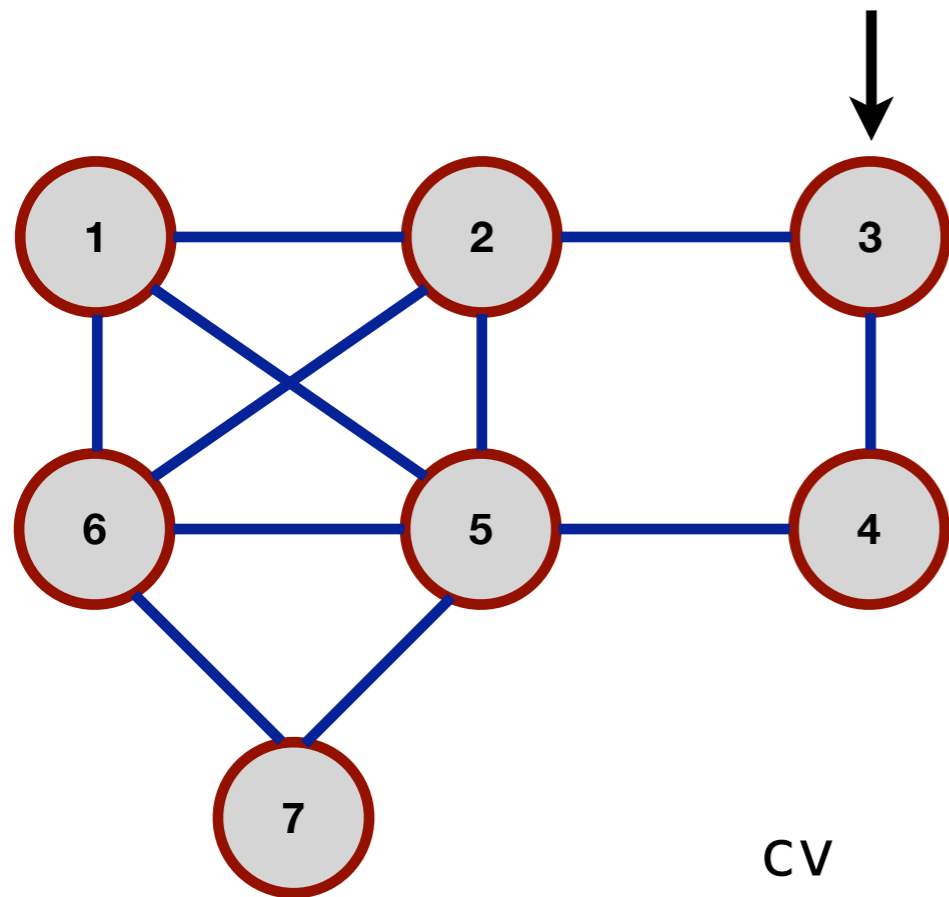
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6

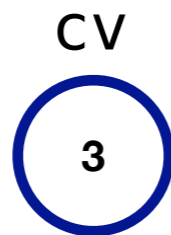


# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



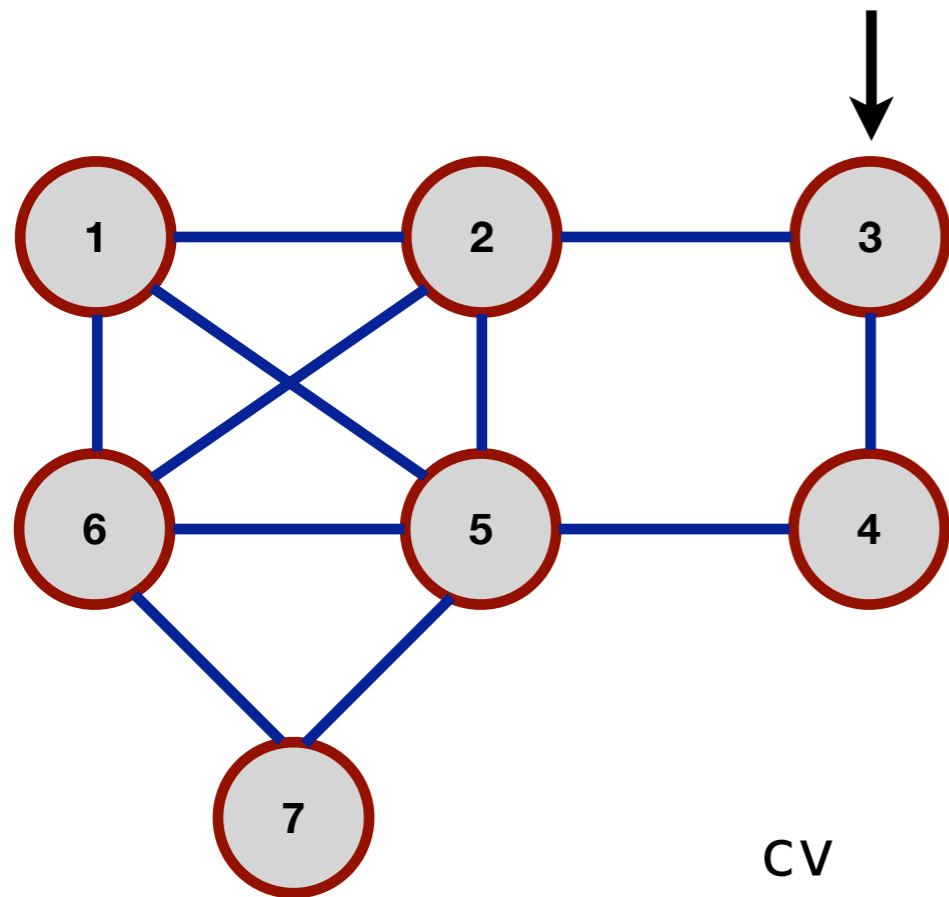
1 2 5 6



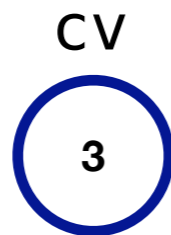
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3

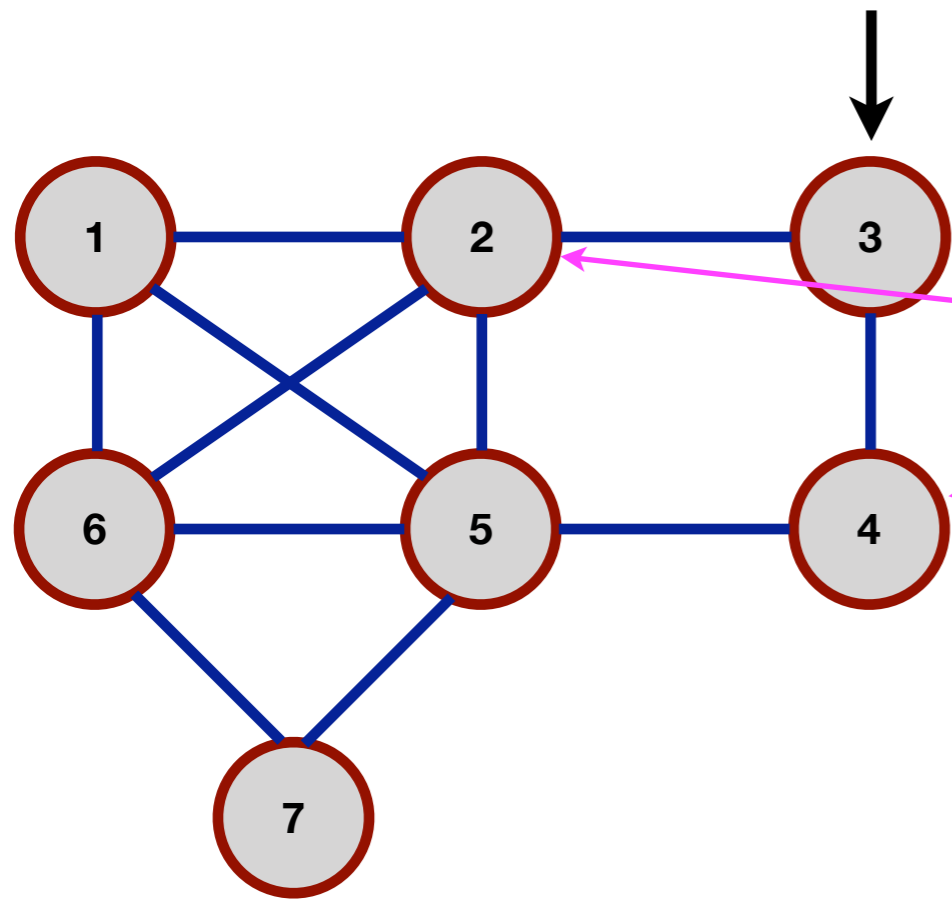


```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

for n in cv.neighbors[]

1 2 5 6 3

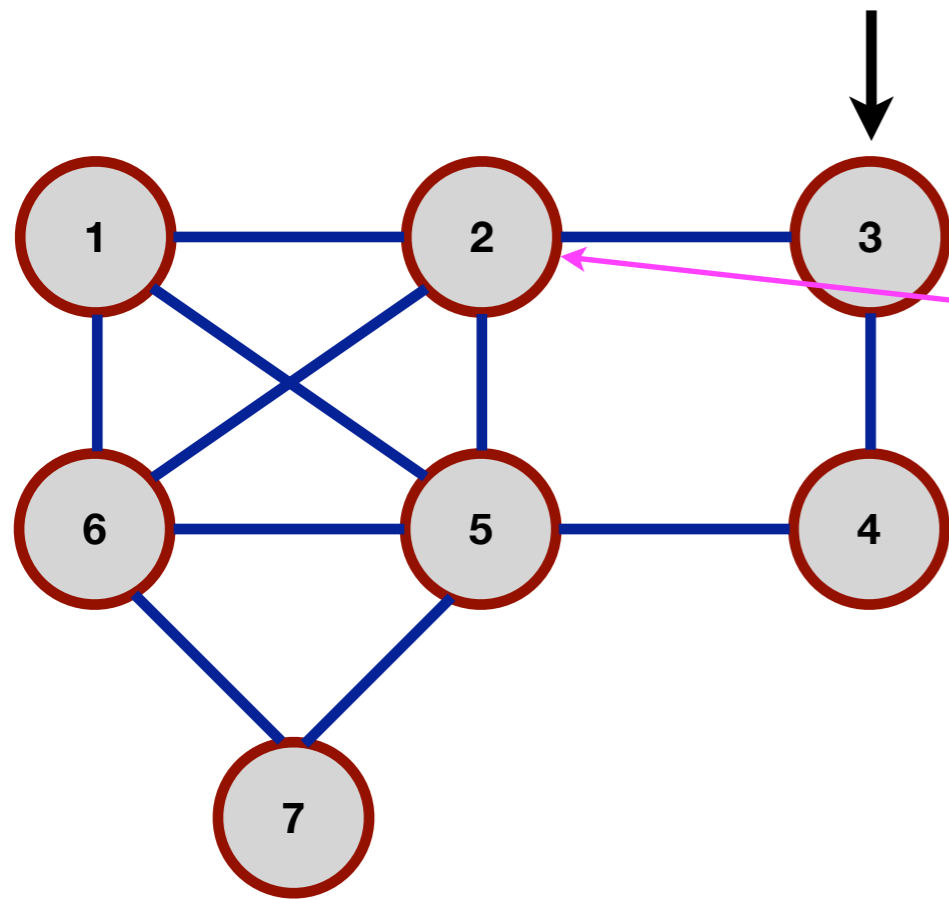


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

if (not n.processed)

1 2 5 6 3

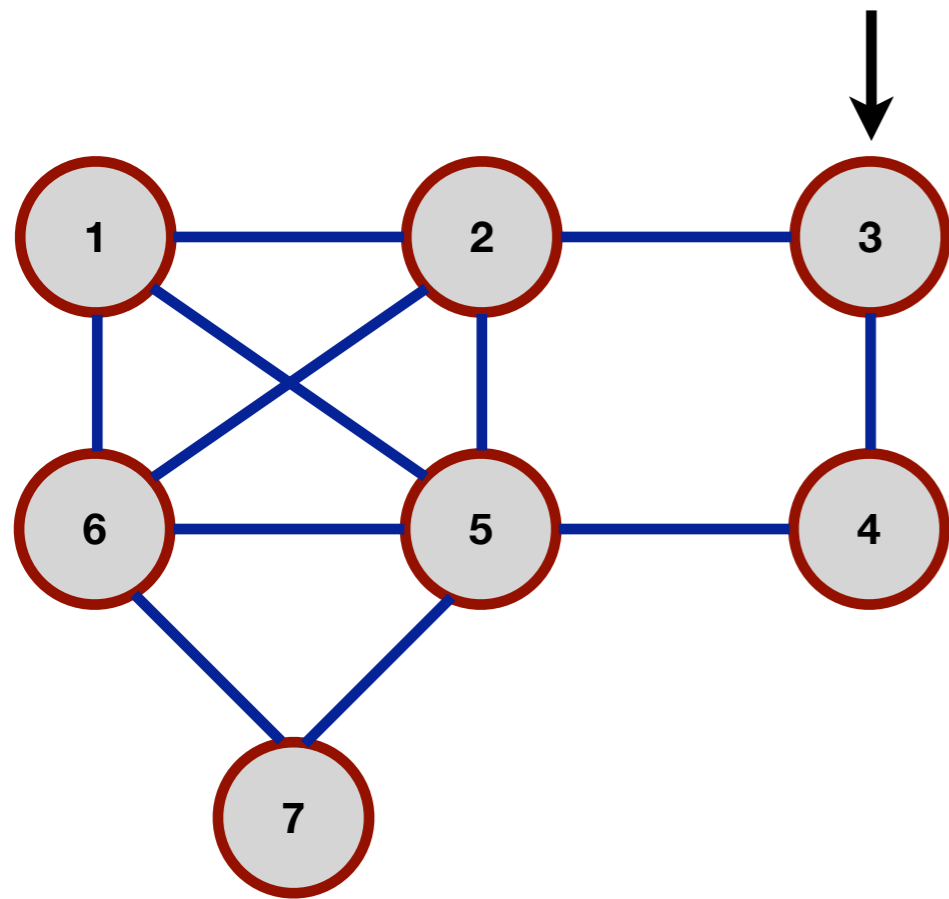


[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3

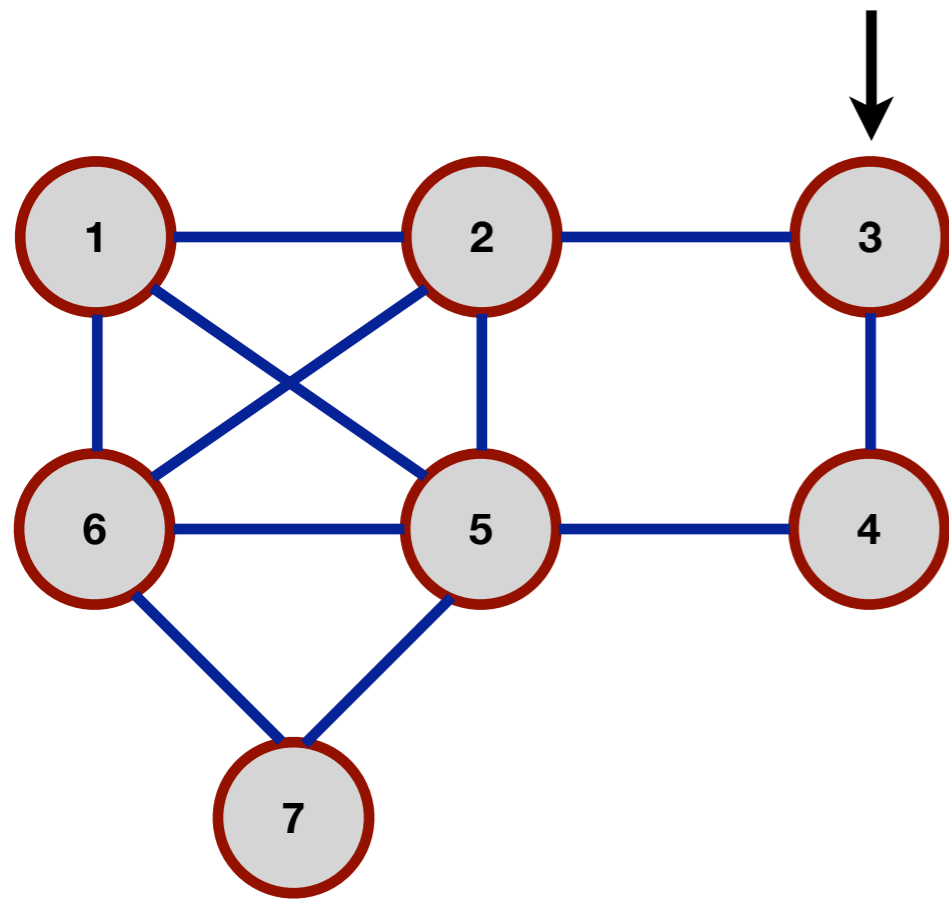


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3

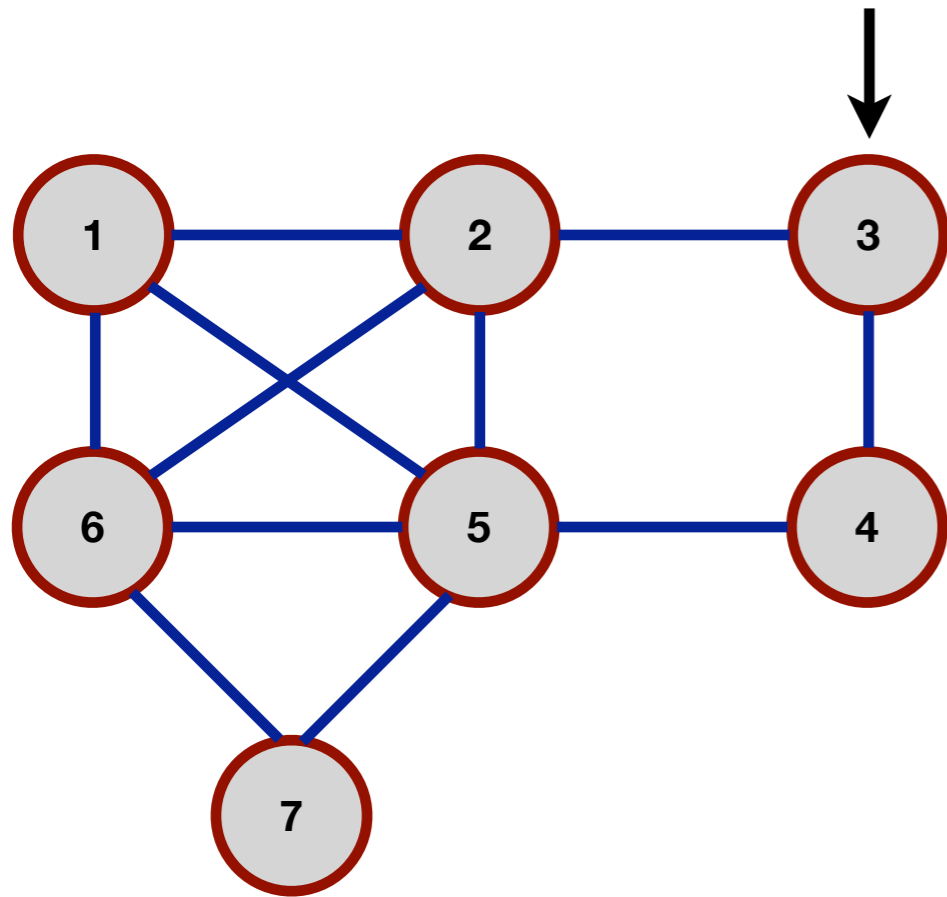


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



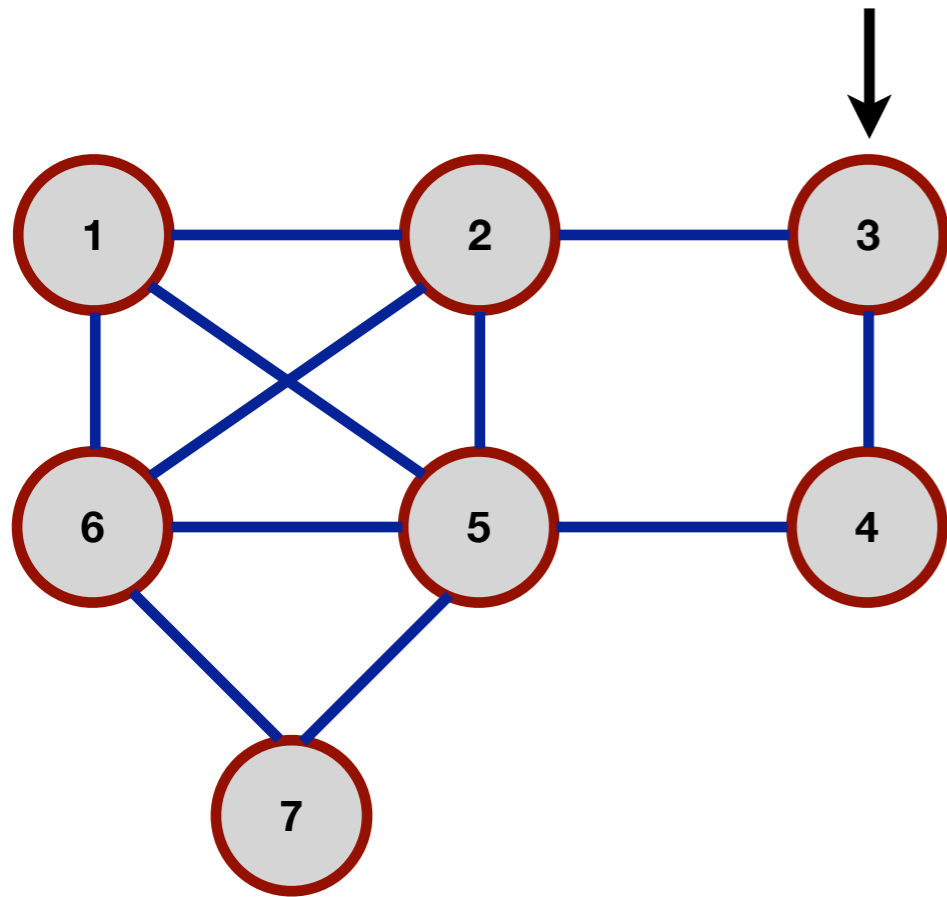
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



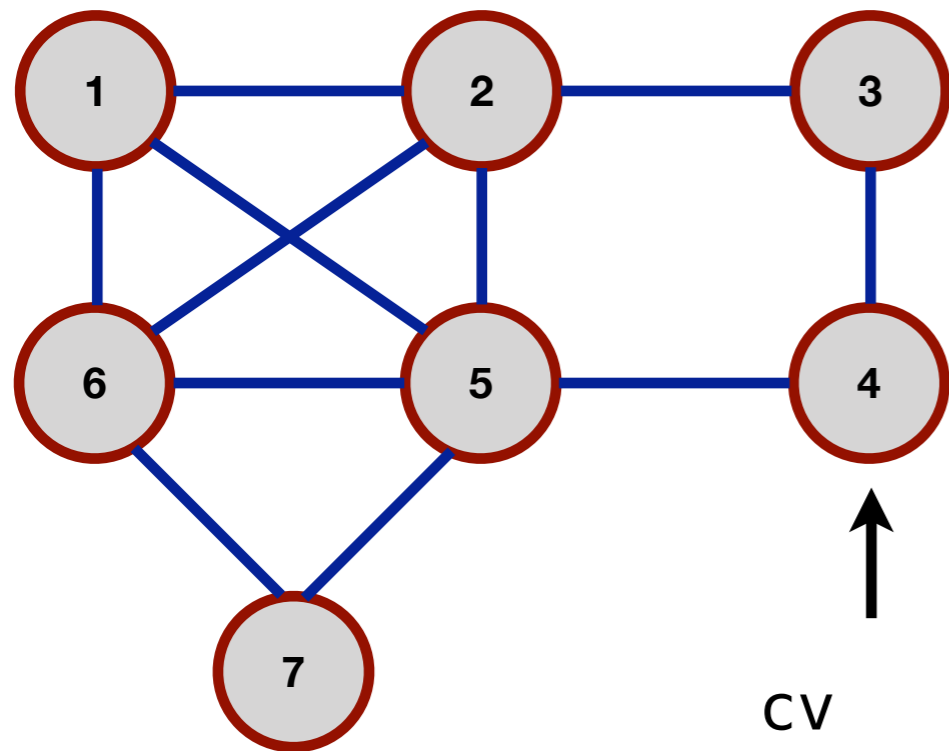
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3

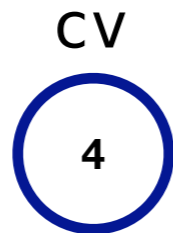


# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



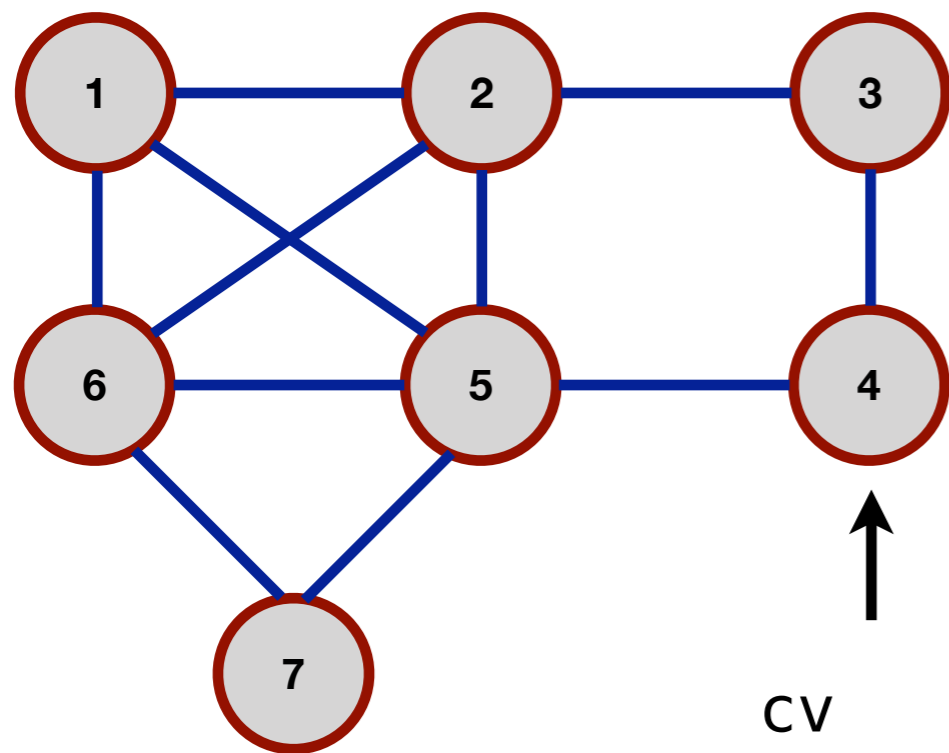
1 2 5 6 3



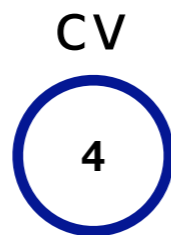
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4

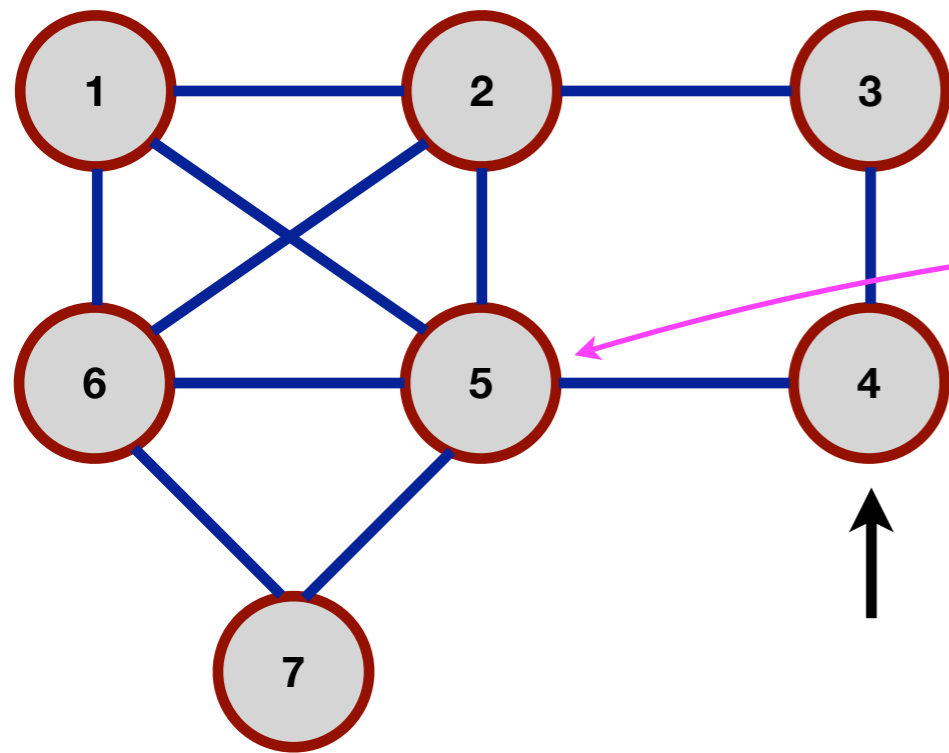


```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4

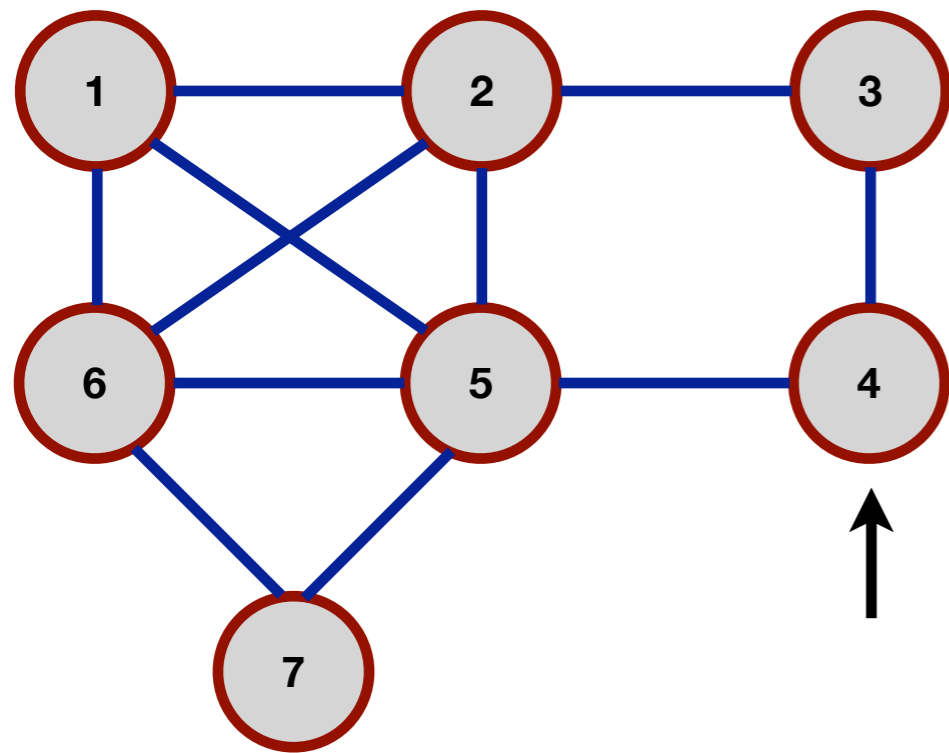


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

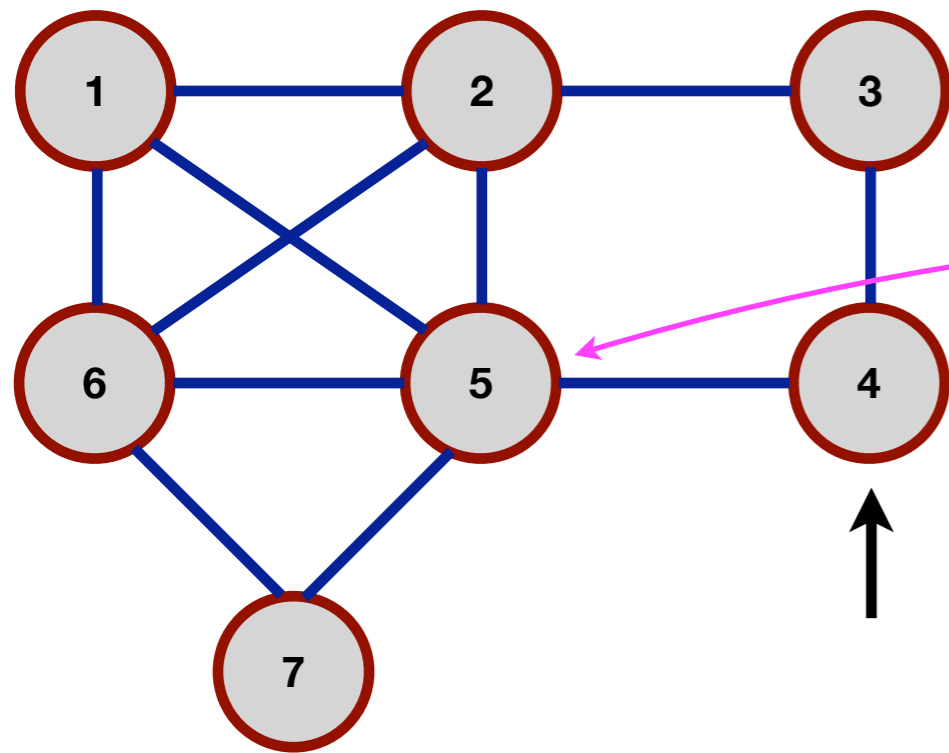
Adjacency List

1 2 5 6 3 4



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4

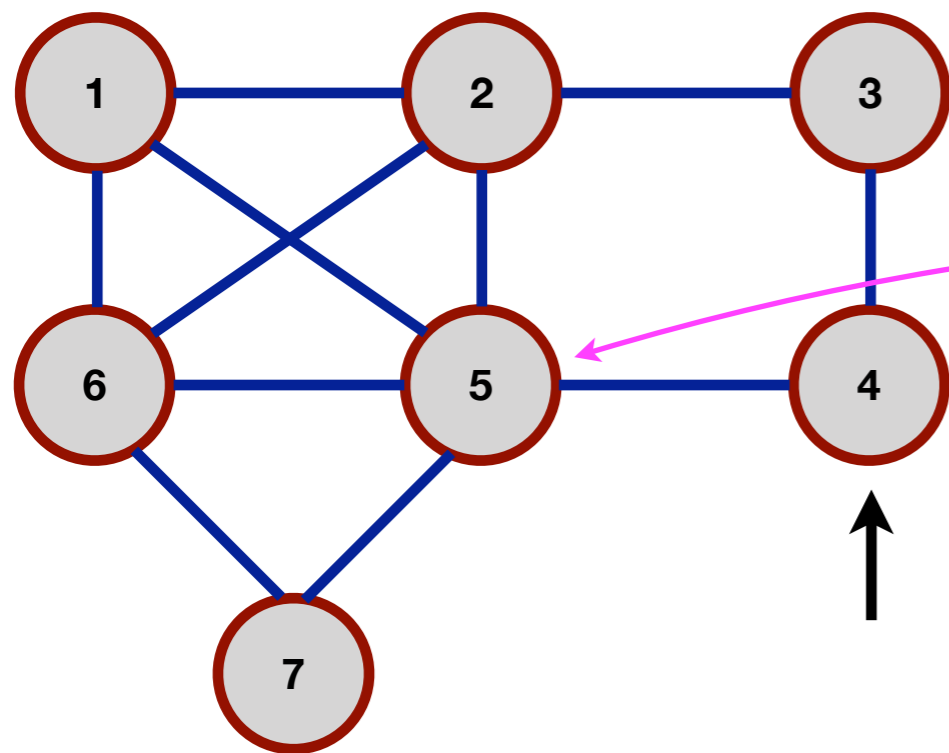


```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```

proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
  
```

endfor

1 2 5 6 3 4

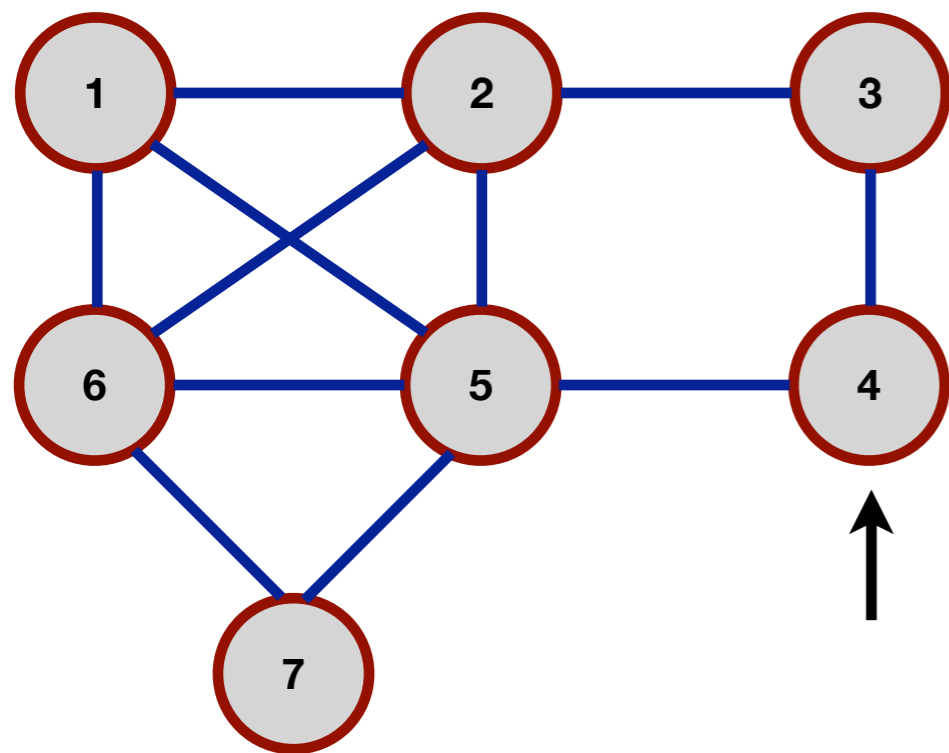


[1]	2	5	6		
[2]	1	3	5	6	
[3]	2	4			
[4]	3	5			
[5]	1	2	4	6	7
[6]	1	2	5	7	
[7]	5	6			

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



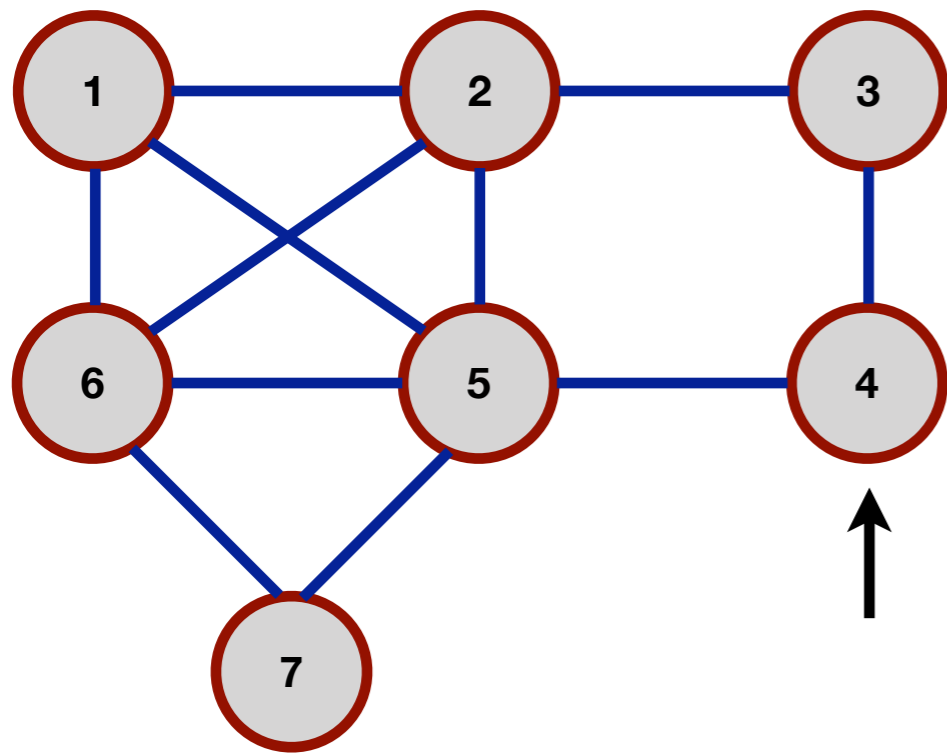
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



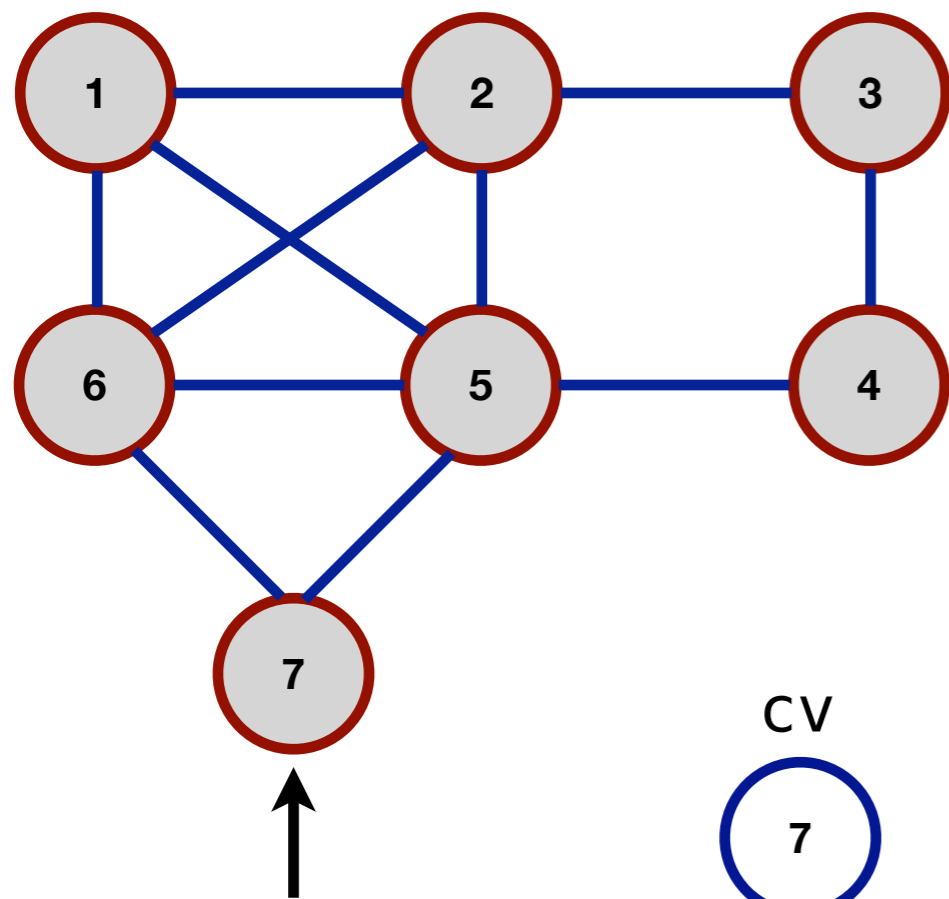
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



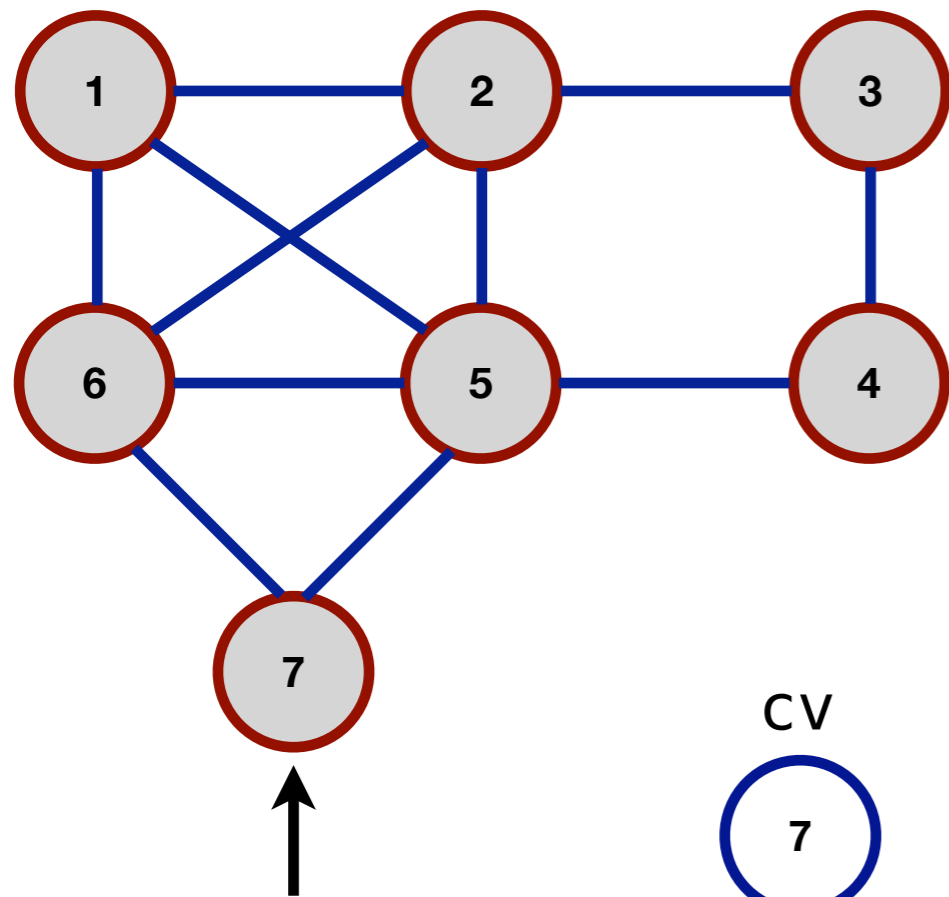
1 2 5 6 3 4



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4 7

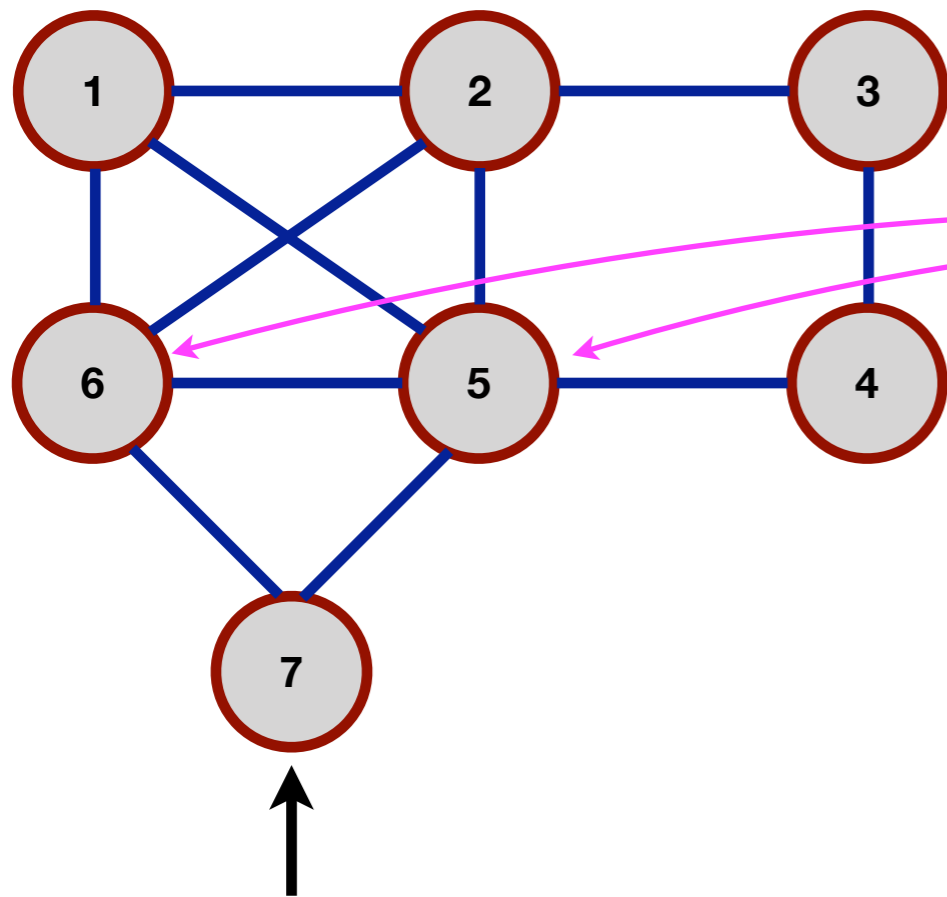


```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4 7



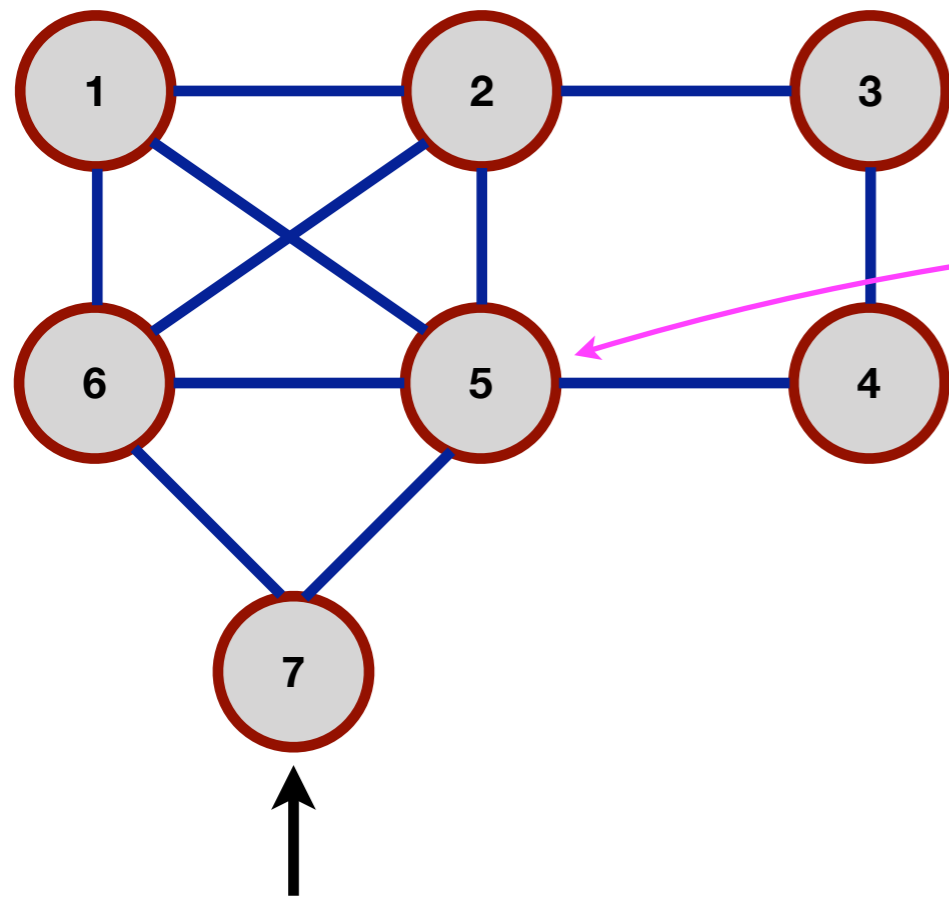
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4 7



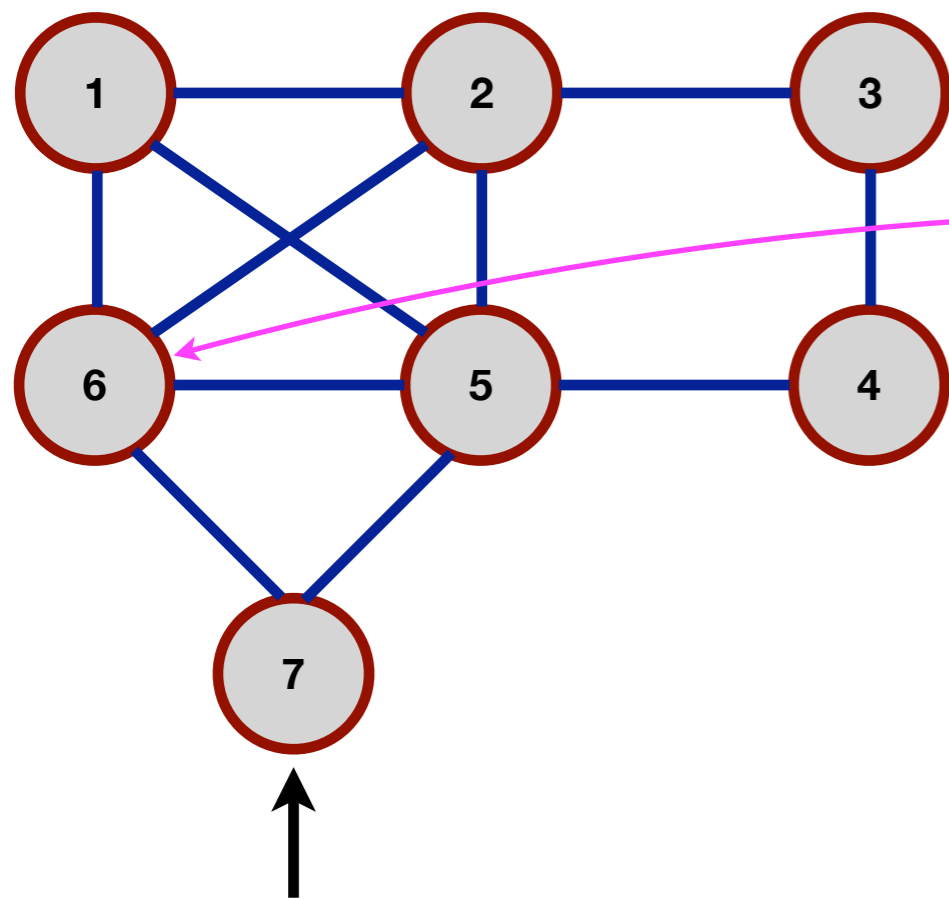
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4 7



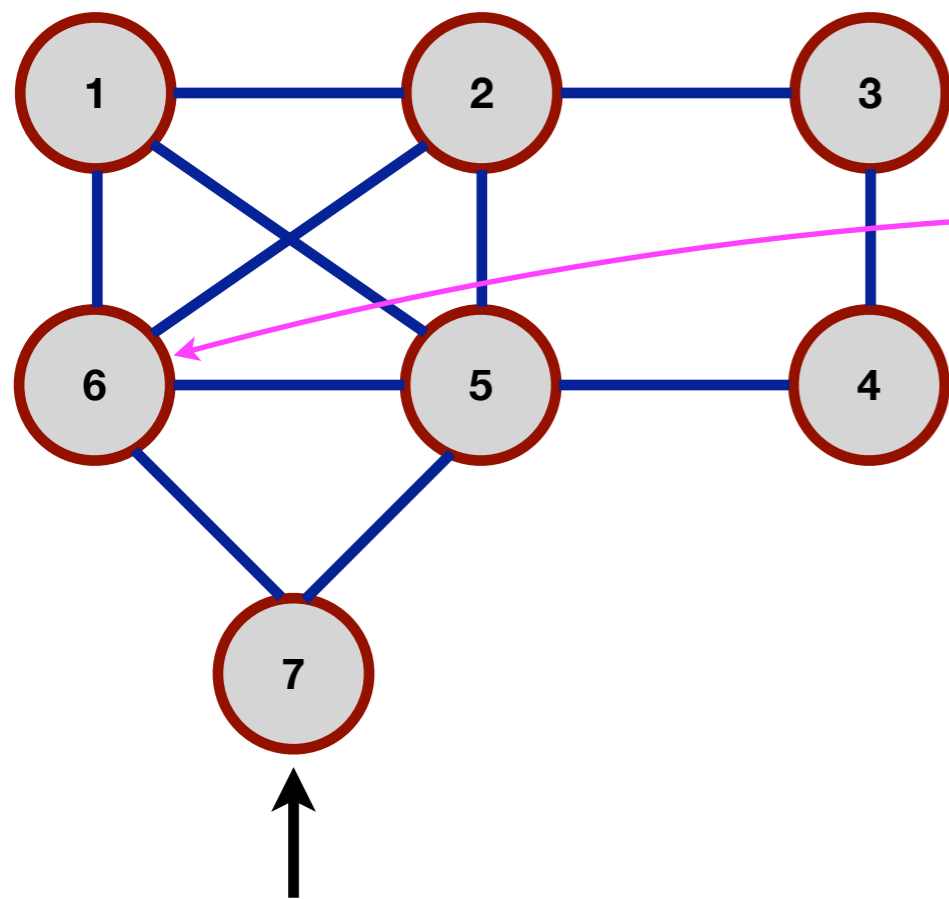
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



1 2 5 6 3 4 7



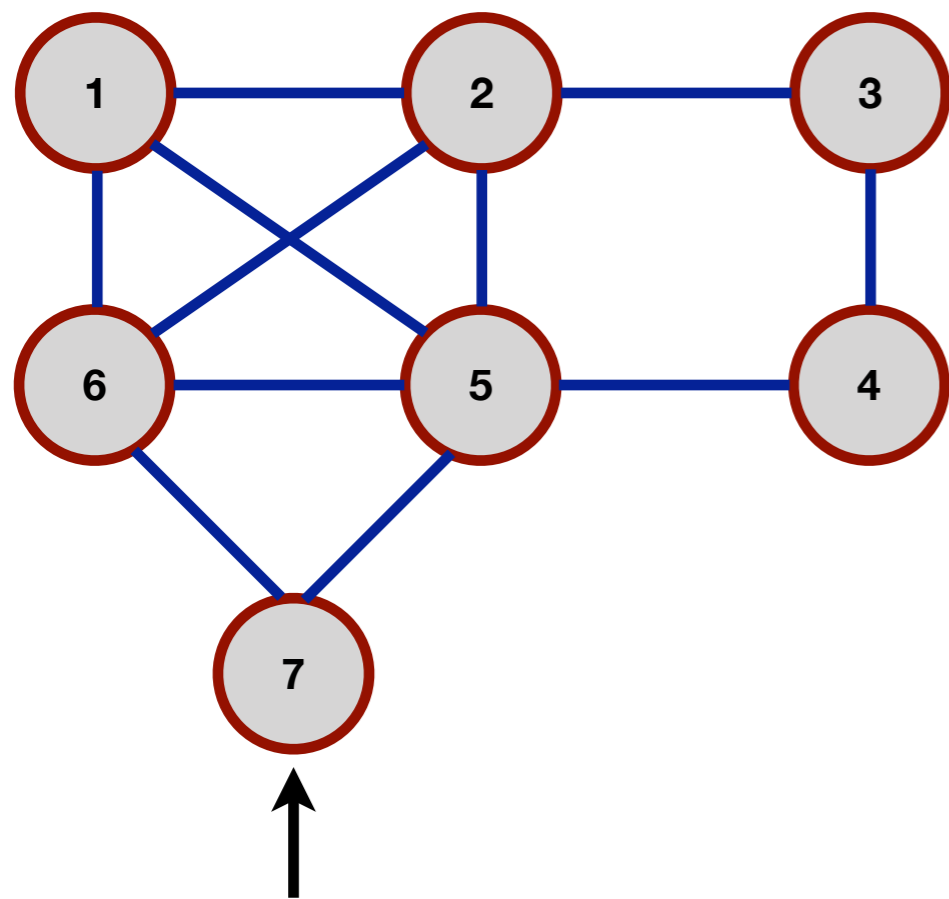
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

```
[1] 2 5 6
[2] 1 3 5 6
[3] 2 4
[4] 3 5
[5] 1 2 4 6 7
[6] 1 2 5 7
[7] 5 6
```

Adjacency List

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



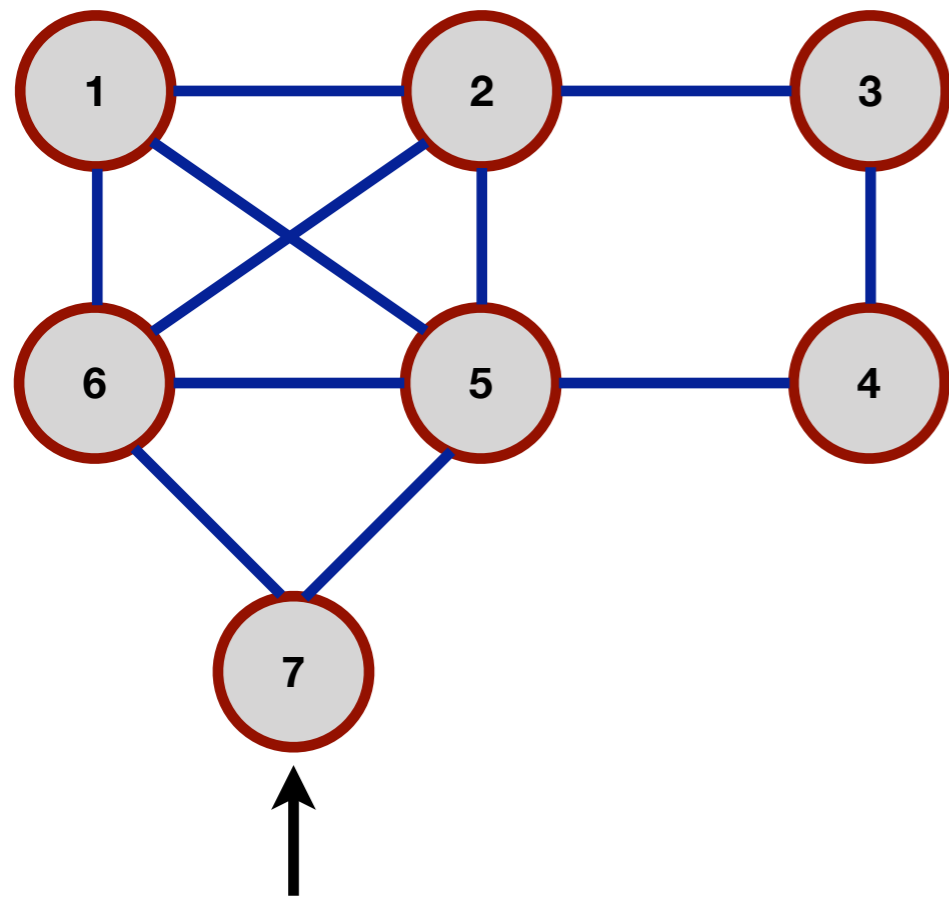
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4 7



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



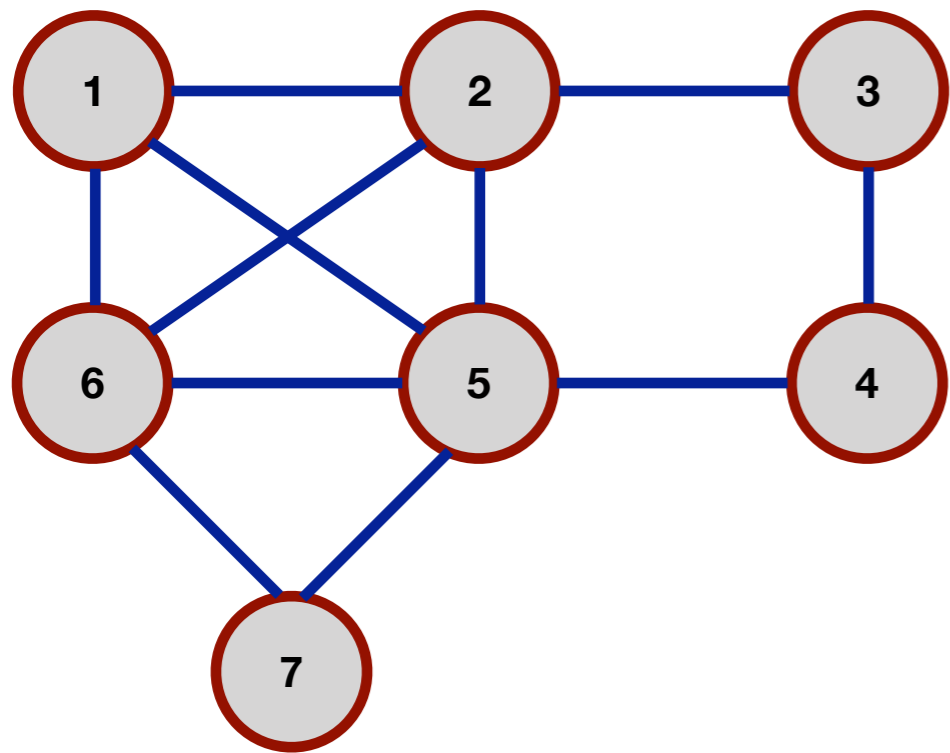
1 2 5 6 3 4 7



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



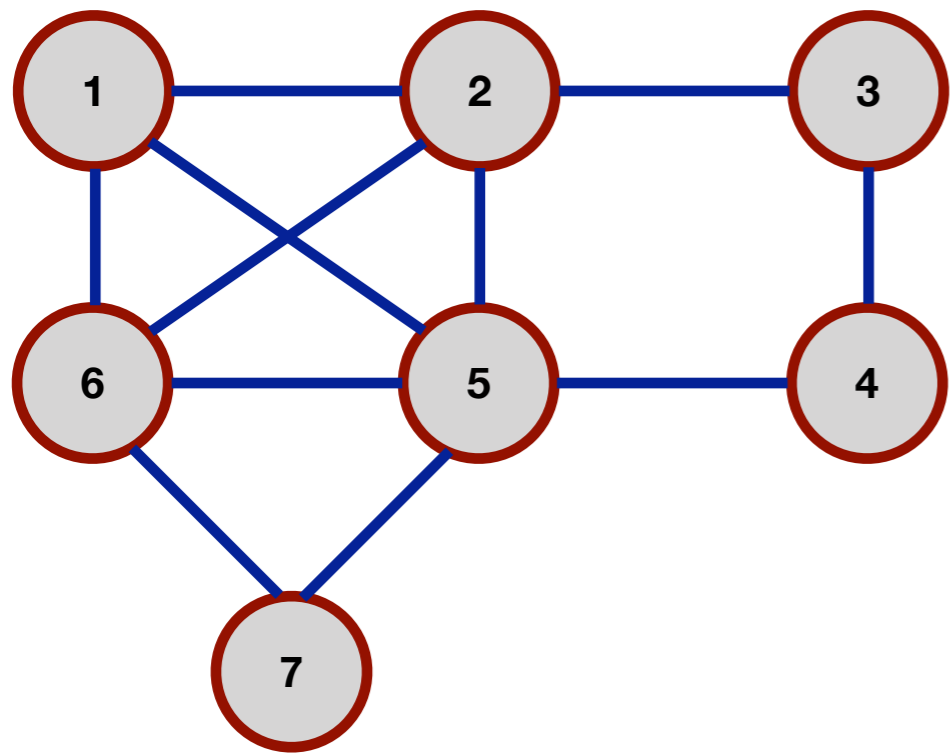
```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4 7



# Breadth-First Search / Traversal

myGraph.BFS(vertex1)



```
proc BFS(fromVertex v)
  Q := new Queue()
  Q.enqueue(v)
  v.processed := true
  while (not Q.empty())
    cv := Q.dequeue()
    print(cv.id)
    for n in cv.neighbors[]
      if (not n.processed)
        Q.enqueue(n)
        n.processed := true
      endif
    endfor
  endwhile
endproc
```

1 2 5 6 3 4 7

