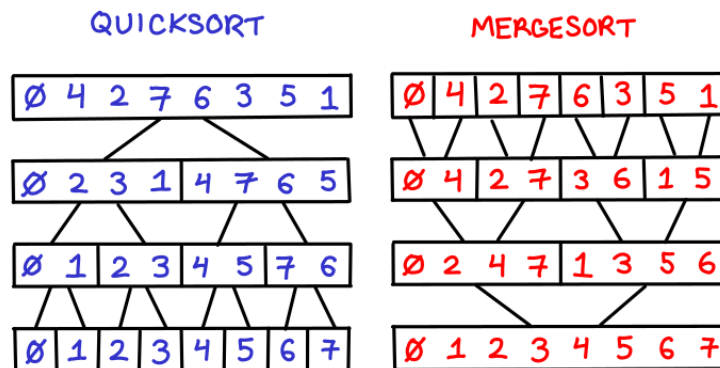


Algorithms

CMPT 435

– Assignment 2 - 100 points

Goals	<ul style="list-style-type: none">to implement a few sorting algorithms and understand their performance.
Requirements and Notes	<ul style="list-style-type: none">Download the the text file <code>magicitems.txt</code> from our web site.Read it line-by-line into an array, reusing (and improving) your code from assignment #1.Develop your own implementation of selection sort, insertion sort, merge sort, and quick sort.Sort using your selection sort. Print the number of comparisons. [10 points]Sort using your insertion sort. Print the number of comparisons. [10 points]Sort using your merge sort. Print the number of comparisons. [30 points]Sort using your quick sort. Print the number of comparisons. [30 points] <p>Shuffle the <code>magicitems</code> array before each sort. Write your own $O(n)$ shuffle routine based on the Knuth shuffle (also known as the Fisher-Yates shuffle, but not known as the Rosanna shuffle, which is different).</p> <ul style="list-style-type: none">Document your code (with line numbers and explanations) and your results (in a table) in a LaTeX document. For the results, note the asymptotic running time of <i>each</i> sort and explain why it is that way. [20 points] <p>Your code must ...</p> <ul style="list-style-type: none">separate structure from presentation.be professionally formatted yet uniquely yours (show some personality) [-∞ if not]use and demonstrate best practices.make me proud to be your teacher.
Resources	<ul style="list-style-type: none">Insertion sort, merge sort, and quick sort are described in our text in sections 2.1, 2.3, and 7.1 respectively.
Submitting Your Work	<p>In addition to your source code, commit your LaTeX document in both <code>.tex</code> and <code>.pdf</code> forms to your GitHub repository. For your code, make many commits to GitHub. If you don't make enough commits, I will not accept your work. Be sure that you make your final commit for this assignment on or before the due date. (See our syllabus for those details.)</p>



Algorithms

CMPT 435

Example Output (Calculating the time is not required, but it's a nice touch. Also note that this output example includes Bubble sort where you're doing Selection sort instead.)

```
n^2      : 443,556.000
n*log(n):  6,246.667
```

Bubble sort

```
Number of comparisons: 221,445
This took              :    7,402 µs
```

Insertion sort

```
Number of comparisons: 109,610
This took              :    2,848 µs
```

Merge sort

```
Number of comparisons: 5,419
This took              :    586 µs
```

Quicksort

```
Number of comparisons: 7,441
This took              :    465 µs
```