Tute 3
Divide and Conquer, Asymptotic Complexity

Topics

- Selection sort revisited
- Asymptotic Analysis
- Recursing recursion
- Quick sort

Selection sort

Sample input: 2, 1, 10, 4, 6

SelectionSort(A, n)

```
1. For i=n-1 downto 1
       a. max = 0
       b. For j=0 to i
            i.   If A[j] > A[max]
                     1. max = j
       c. Swap A[max] and A[i]
```

Refer to my git Repo: https://goo.gl/K9SddV

Asymptotic Analysis

1. What is Big Oh
2. Sample graph for running time of *selection sort* and how it is O(n^2)
3. Mathematical concept - not just for measuring the running time of algorithms.

Recursing recursion

1. Refer to recursive sum program
2. Recursion stack. Understand how the values are returned from recursion calls.

Quicksort: The strategy

1. Choose a pivot *p*
2. Place the pivot in it's final position
    a. Move elements less than *p* to the left of the array
    b. Move elements greater than *p* to the right of the array
    c. Place *p* in it's final position
3. Recurse on the left and right subarrays of *p*

Sample input: 8, 10, 2, 1, 4, 2, 3

Analysis of Quicksort

1. Divide and conquer
2. Best case, worst case and average case