CSE 5311.006
ADHIRAJ SEN (1002264465)
AXS4465

HW 3

Ex 3.1-4

Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

2^{n+1} can be written as $2^n \times 2$

If $2^{n+1} \leq c \cdot 2^n$, for $n \geq n_0$
Substituting,

$$2 \times 2^n \leq c \cdot 2^n$$

Divide by 2^n,

$$2 \leq c$$

Since, $c = 2$ holds for $n \geq 0$

$2^{n+1} = O(2^n)$, TRUE

2^{2n} can be written as $2^{2n} = (2^n)^2$

For, $2^{2n} \leq c \times 2^n$ for, $n \geq n_0$

$\Rightarrow (2^n)^2 \leq c \times 2^n$
Dividing by 2^n,

$$2^n \leq c$$

This is not true since 2^n grows exponentially and will surpass constant c.

$$2^{2n} \neq O(2^n), \text{ FALSE}$$

1) Find the runtime for
a) Best Case
b) Worst Case

Ans 1 Best Case :

a) - The outer loop runs from $j=2$, till $n-1$ iteratio
- The while loop condition $A[i] > key$ is never true because array is sorted.
- Each key is inserted without shifting positions

$$(n-1) = O(n)$$

Worst case :

- The outer loop runs from $j=2$, given n-literation
- While loop runs for every previous statement
- For each j, number of comparisons are $j-1$

Total comparisons,

$$\sum_{j=2}^{n}(j-1) = 1+2+3 \cdots + (n-1)$$

$$\text{sum formula} \rightarrow \frac{(n-1)n}{2} = \Theta(n^2)$$

- each comparison results in a shift.
- each element is shifted everytim

no. of shifts, $\Theta(n^2)$

2) For Bubble Sort what is
a) worst case time complexity and why?
b) best case time complexity and why?

Ans 2

a) Worst case would be a reverse sorted array

$$\text{Comparisons}, \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} = O(n^2)$$

Worst case is $O(n^2)$ because every element must be compared and swapped multiple times in an unsorted list

b) Best case would be already sorted array
In a sorted array, swap flag prevents further iterations making bubble sort a one pass through the array.

Best case is $O(n)$ when array is alread sorted and an optimization (swap flag) is used

3) For selection sort what is:
a) Worst case time complexity and Why?
b) Best " " " " "

Ans 3

a) Worst case :
Regardless of input order, Selection Sort performs the same number of comparisons because it scans entire remaining unsorted portion to find min.

$$\text{comparisons}, \sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2} = O(n^2)$$

no. of swaps at most n-1 swaps is $O(n)$

Dominant term and Worst case is $O(n^2)$

b) Selection Sort doesn't have early termination
" " still scans entire remaining unsorted portion to find min element

$$\text{comparisons}, \sum_{i=1}^{n-1}(n-i) = O(n^2)$$

4 Find the runtime complexity using Θ-notation of

a) $T(n) = \log n + n + 1$

Ans $\Theta(n)$

b) $T(n) = \log n + \log n + 10^{1000}$

Ans $\Theta(\log n)$

c) $T(n) = n \log n + \log n + n + 3$

Ans $\Theta(n \log n)$

d) $T(n) = 2^n + n!$

Ans $\Theta(n!)$

5 Prove the order of growth:

a) $\frac{1}{2} n (n-1) = \Theta(n^2)$

Ans $\frac{1}{2}n^2 - \frac{1}{2}n$

Limits, $\lim_{n \to \infty} \frac{\frac{1}{2}n(n-1)}{n^2} = \lim_{n \to \infty} \frac{\frac{1}{2}(n^2-n)}{n^2} = \lim_{n \to \infty} \frac{1}{2}\left(1 - \frac{1}{n}\right)$

Since limit is constant,

$$\frac{1}{2} n(n-1) = \Theta(n^2)$$

b) $\log n \in o(\sqrt{n})$

Ans $\lim_{n \to \infty} \frac{\log n}{\sqrt{n}} = 0$

$\implies \log n = o(\sqrt{n})$

c) $\log n = O(\sqrt{n})$

Ans $\lim_{n \to \infty} \frac{\log n}{\sqrt{n}}$

$\implies \lim_{n \to \infty} \frac{\log n}{\sqrt{n}} = 0$

$\implies \log n = O(\sqrt{n})$

d) $n! = \Omega(2^n)$

Ans $\lim_{n \to \infty} \frac{n!}{2^n}$

$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

$\lim_{n \to \infty} \frac{n!}{2^n} = \lim_{n \to \infty} \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{2^n} = \lim_{n \to \infty} \sqrt{2\pi n}\left(\frac{n}{2e}\right)^n$

$n! = \Omega(2^n)$

e) $2^n = O(n^3)$

$$\lim_{n \to \infty} \frac{2^n}{n^3}$$

$$\implies \lim_{n \to \infty} \frac{2^n}{n^3} = \infty$$

limit is infinite, $2^n \neq O(n^3)$

f) $4n^3 \in \Omega(n)$

$$\lim_{n \to \infty} \frac{4n^3}{n}$$

$$\lim_{n \to \infty} 4n^2 = \infty$$

$4n^3 = \Omega(n)$