Name: Harshit Jain
Access ID: hmj5262
Recitation: 8

Problem 0 **Points:**

Acknowledgements

(a) I did not work in a group.

(b) I did not consult without anyone my group members.

(c) I did not consult any non-class materials.

Problem 1

Points:

Analyze Running Time

1. **Runtime:** $\theta(n^2)$

   When $i = 1$, then $j = 1$. Here, while loop runs for $\frac{n-1}{5}$ times.

   When $i = 2$, then $j = 2$. Here, while loop runs for $\frac{n-2}{5}$ times.

   $\vdots$

   When $i = (n-5)$, then $j = (n-5)$. Here, while loop runs for 1 time.

   $\vdots$

   When $i = (n-1)$, then $j = (n-1)$. Here, while loop runs for 1 time.

   Total time: $\frac{n-1}{5} + \frac{n-2}{5} + \cdots + \frac{5}{5} + 1 + 1 + 1 + 1 + 1 \Rightarrow \theta(n^2)$

2. **Runtime:** $\theta(n^2)$

   When $i = 1$, then while loop runs from 4 to n. Time $= (n-4)$

   When $i = 2$, then while loop runs from 8 to n. Time $= (n-8)$

   $\vdots$

   When $i = \frac{n}{4}$, then while loop will run 1 time. Time $= 1$

   For loop will run for total of n times.

   $\Rightarrow \theta(n^2)$

3. **Runtime:** $\theta(nlogn)$

   The main while loop will run for $\theta(logn)$ time as i is reduced to its half after every iteration.

   The inner for loop will run for $\frac{n}{2} + \frac{n}{2^2} + \frac{n}{2^3} + \cdots + 1 = \theta(n)$

   So, total time $\Rightarrow \theta(nlogn)$

Problem 2

Points:

Polynomials and Horner's rule

$$P(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_nx^n; x = x_0$$

(a) For Addition: The Brute Force Algorithm will perform **n** additions in total.
So, Time taken for additions $= O(n)$
For Multiplication: $P(x) = a_0 + [a_1 \cdot x_0] + [a_2 \cdot x_2 \cdot x_2] + \cdots + [a_n \cdot x_n \cdot x_n \cdots (ntimes)]$
So, Time taken $= T_n = 1 + 2 + 3 + \cdots + n$
$= \frac{n(n+1)}{2}$
$= \frac{n^2}{2} + \frac{n}{2}$
$= O(n^2)$
Therefore, total time complexity for Brute Force Algorithm $= O(n^2) + O(n) = O(n^2)$

(b)

(c)

Problem 3

Points:

Solving recurrences

(a) Height $= \log_2 n$

Branching factor $= 2$

The size of the sub-problems at depth $k = \frac{n}{2^k}$

Number of sub-problems at depth $k = 2^k$

Total work done $=$

$$\sum_{k=0}^{\log_2 n} (2^k) \cdot (\frac{n}{2^k})^{\frac{1}{2}} \Rightarrow \sqrt{n} \cdot \sum_{k=0}^{\log_2 n} (2^{\frac{k}{2}}) \Rightarrow \sqrt{n} \cdot \theta(\sqrt{n}) \Rightarrow \theta(n)$$

(Note that, $\sum_{k=0}^{\log_2 n} (2^{\frac{k}{2}})$ is a geometric series $= 2^{\frac{0}{2}} + 2^{\frac{1}{2}} + \cdots + 2^{\frac{\log_2 n}{2}} = \theta(\sqrt{n})$)

(b) Height $= \log_3 n$

Branching factor $= 2$

The size of the sub-problems at depth $k = \frac{n}{3^k}$

Number of sub-problems at depth $k = 2^k$

Total work done $=$

$$\sum_{k=0}^{\log_3 n} (2^k) \cdot (\frac{n}{3^k})^0 \Rightarrow \sum_{k=0}^{\log_3 n} (2^k) \Rightarrow 2^0 + 2^1 + 2^3 + \cdots + 2^{\log_3 n} \Rightarrow \theta(n^{\log_2 3})$$

(c) Height $= \log_4 n$

Branching factor $= 5$

The size of the sub-problems at depth $k = \frac{n}{4^k}$

Number of sub-problems at depth $k = 5^k$

Total work done $=$

$$\sum_{k=0}^{\log_4 n} (5^k) \cdot (\frac{n}{4^k})^1 \Rightarrow n \cdot \sum_{k=0}^{\log_4 n} ((\frac{5}{4})^k) \Rightarrow n \cdot ((\frac{5}{4})^0 + (\frac{5}{4})^1 + \cdots + (\frac{5}{4})^{\log_4 n}) \Rightarrow n \cdot \theta(\frac{5^{\log_4 n}}{n}) \Rightarrow \theta(n^{log_5 4})$$

(d) Height $= \log_7 n$

Branching factor $= 7$

The size of the sub-problems at depth $k = \frac{n}{7^k}$

Number of sub-problems at depth $k = 7^k$

Total work done $=$

$$\sum_{k=0}^{\log_7 n} (7^k) \cdot (\frac{n}{7^k})^1 \Rightarrow n \cdot \sum_{k=0}^{\log_7 n} 1 \Rightarrow n \cdot \log n \Rightarrow \theta(n \log n)$$

(e) Height $= \log_3 n$

Branching factor $= 9$

The size of the sub-problems at depth $k = \frac{n}{3^k}$

Number of sub-problems at depth $k = 9^k$

Total work done $=$

$$\sum_{k=0}^{\log_3 n} (9^k) \cdot (\frac{n}{3^k})^2 \Rightarrow n^2 \cdot \sum_{k=0}^{\log_3 n} 1 \Rightarrow n^2 \cdot \log n \Rightarrow \theta(n^2 \log n)$$