Name: Grishma Maheshbhai Thumar

Student Id: 922950012

Class: CSC 510, Spring 2025

Project: Homework 2

1a) Answer

1) Method of recursion trees.

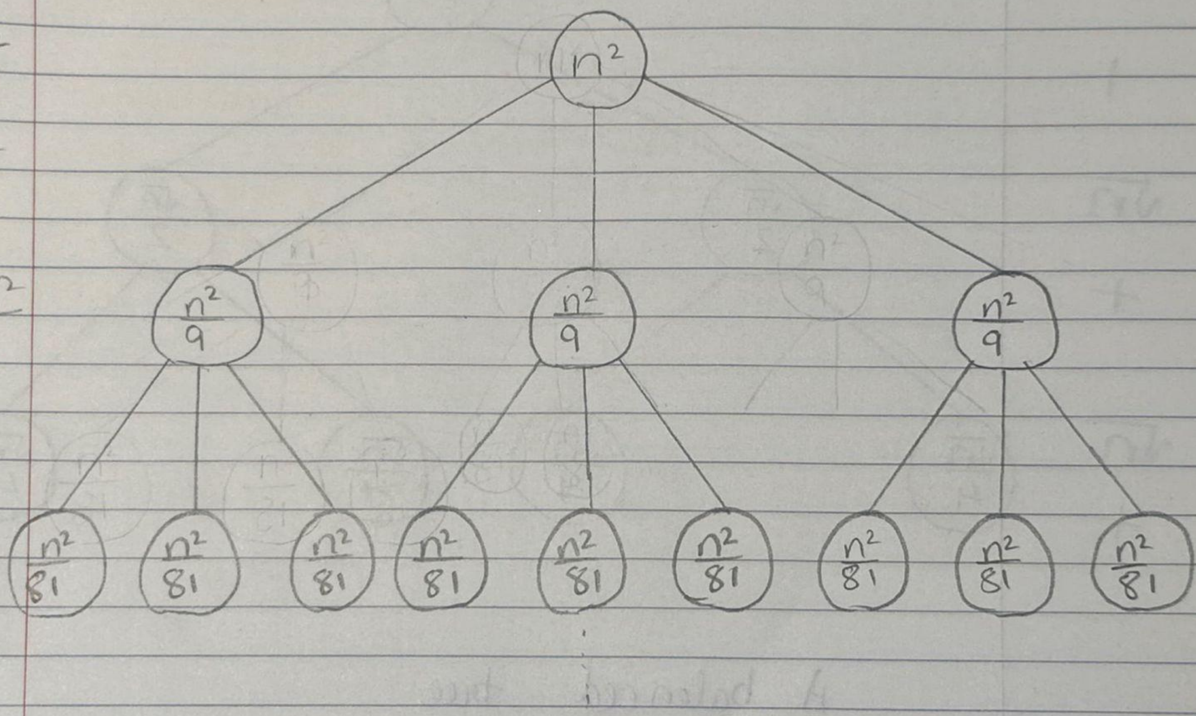
a) $A(n) = 3A(n/3) + O(n^2)$

Root $= n^2$

Children = each child $(n/3)^2$

Grand children = each grand child $(n/9)^2$

$n^2 + \frac{3n^2}{9} + \frac{9n^2}{81} + \cdots$

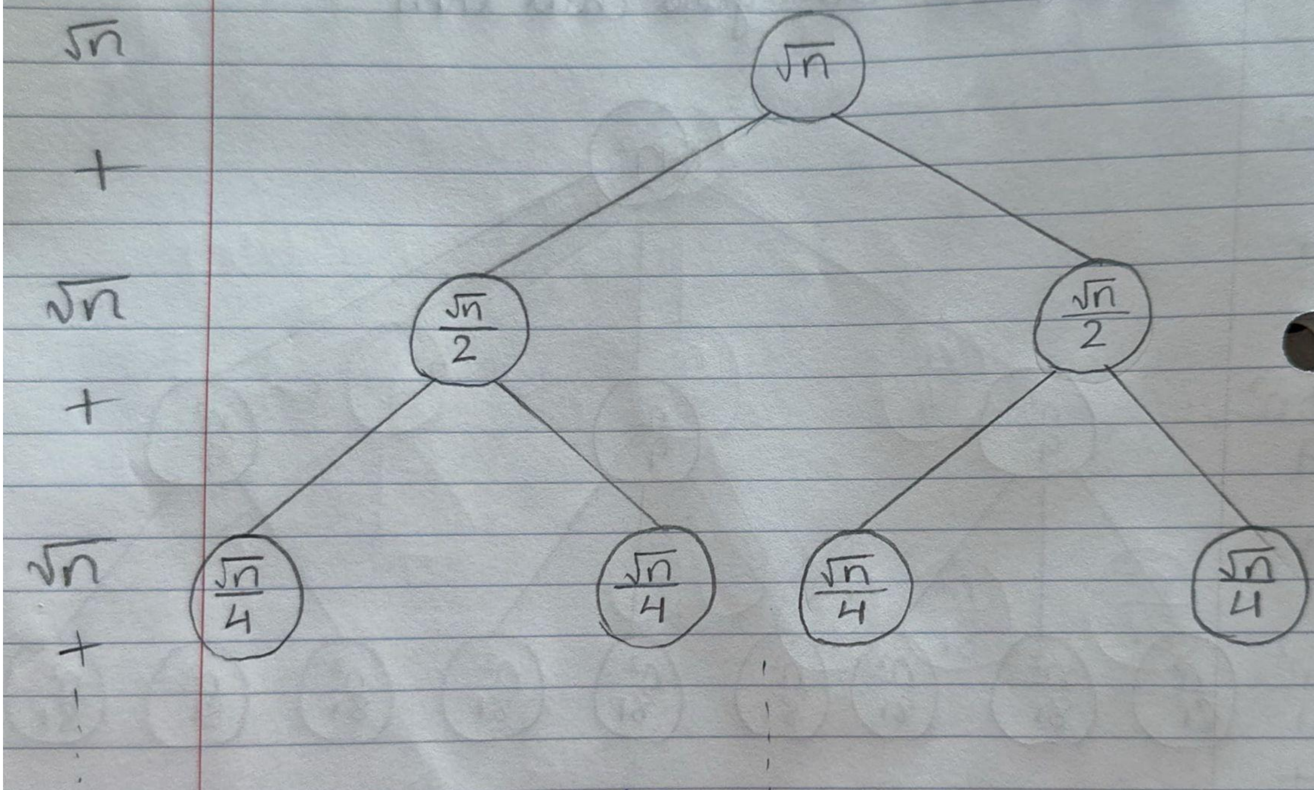


A root heavy tree, so $A(n) = O(n^2)$

1b) Answer

b) $B(n) = 2B(n/4) + O(\sqrt{n})$

Root $= \sqrt{n}$
Children $=$ each child $\sqrt{n}/2$
Grand children $=$ each grandchild $\sqrt{n}/4$



A balanced tree as the sum converges to $O(\sqrt{n})$. So, final complexity is $B(n) = O(\sqrt{n})$

2. c) $C(n) = 4\,C(n/13) + \Theta(1)$

$r = 4$; $c = 13$; $b = \log_c r = \log_{13}(4)$; $a = 0$ (as it is $\Theta(1^0)$)

Since $a < b$, by Master Theorem

$C(n) = \Theta(n \wedge (\log_{13}(4)))$

2. d) $D(n) = 9\ D(n/3) + \Theta\ (n^2)$

$r = 9; c = 3; b = \log_3 (9) = 2; a = 2$

Since a = b, by Master Theorem

$D(n) = \Theta\ (n^2 \log(n))$

2. e) $E(n) = 19\ E(n/2) + \Theta\ (n^4)$

$r = 19; c = 2; b = \log_2 (19); a = 4$

Since a > b, by Master Theorem

$E(n) = \Theta\ (n^4)$

2. f) $F(n) = 2\ F(n/8) + \Theta\ (\sqrt[3]{n})$

$r = 2; c = 8; b = \log_8 (2) = 1/3; a = 1/3$

Since a = b, by Master Theorem

$F(n) = \Theta\ (n^{1/3} \log(n))$

2. g) $G(n) = 6\ G(n/7) + \Theta\ (n)$

$r = 6; c = 7; b = \log_7 (6); a = 1$

Since a > b, by Master Theorem

$G(n) = \Theta\ (n)$

3. Pokémon Cards

a) Brute Force Algorithm

```
CARDBRUTE (P[1, …., n]) :
    max_profit = 0
    for i = 1 to n:
        for j = i to n:
            profit = P[j] – P[i]
            if profit > max_profit:
                max_profit = profit
    return max_profit
```

Time complexity is $\Theta\ (n^2)$ since there are 2 nested for loops.

b) Divide and Conquer Algorithm

```
CARDDNC (P[1, …, n]) :
    if n == 1:
        return 0
    mid = n / 2
    left_profit = CARDDNC (P [1, …, mid])
    right_profit = CARDDNC (P [mid+1, …, n])
    min_left = min (P [1, …., mid])
    max_right = max (P [mid+1, … , n])
    cross_profit = max_right – min_left
    return max (left_profit, right_profit, cross_profit)
```

Time complexity is Θ (n log(n)) due to recurrence $T(n) = 2T(n/2) + O(n)$

4. Karatsuba's algorithm

$X = 10^{2n/3}A + 10^{n/3}B + C$

$Y = 10^{2n/3}D + 10^{n/3}E + F$

Multiplying X and Y we get,

$XY = (10^{2n/3}A + 10^{n/3}B + C)\ (10^{2n/3}D + 10^{n/3}E + F)$

Expanding using the distributive property:

$$XY = 10^{4n/3}AD + 10^{3n/3}(AE+BD) + 10^{2n/3}(AF+BE+CD) +10^{n/3}(BF+CE) + CF$$

a) Thus, the coefficients are:

a. $w_4 = AD$
b. $w_3 = AE + BD$
c. $w_2 = AF + BE + CD$
d. $w_1 = BF + CE$
e. $w_0 = CF$

b) We are given that

$w_2 = (-2M_1 + M_2 + M_3 – 2M_5) / 2$

From part a we have $w_2 = AF + BE + CD$

We know that,

$M_1 = AD$

$M_2 = (A + B + C) * (D + E + F)$

$= AD + AE + AF + BD + BE + BF + CD + CE + CF$

$M_3 = (A – B + C) * (D – E + F)$

$= AD – AE + AF – BD + BE – BF + CD – CE + CF$

$M_5 = CF$

Now,

$(-2M_1 – 2M_5) = -2AD – 2CF$

$M_2 + M_3 = 2AD + 2AF + 2BE + 2CD + 2CF$

So, from the equation that we have

$w_2 = (-2M_1 + M_2 + M_3 – 2M_5) / 2$

$= (-2AD – 2CF + 2AD + 2AF + 2BE + 2CD + 2CF) / 2$

$= (2AF + 2 BE + 2CD) / 2$

$= AF + BE + CD$

Which is equal to the one we got in part (a).

Thus, the equation of w_2 is correct by comparing it to my answer in part (a).

c) M_1 , … , M_5 takes 5 recursive multiplications of two n/3 – digit numbers.

Multiplying and dividing by constants are O(n) operations.

Thus, the recurrence for time T(n) is

$T(n) = 5T(n/3) + O(n)$

By Master Theorem,

$r = 5; c = 3; b = \log_3 (5) = 1.465; a = 1$

Since, $a < b$

That is, $1 < 1.465$

So, $T(n) = O(n^b) = O(n^{1.464})$

In Karatsuba's algorithm,

$r = 2; c = 3; b = \log_2 (3) = 1.585; a = 1$

Since, $a < b$

So, Time complexity is: $O(n^b) = O(n^{1.585})$

Since, $1.465 < 1.585$ thud $O(n^{1.465})$ grows slower than $O(n^{1.585})$ this algorithm (three-way split variant) is faster compared to Karatsuba's algorithm.