Name: Grishma Maheshbhai Thumar

Student Id: 922950012

Class: CSC 510, Spring 2025

Project: Homework 1

1. A) $7n^2 + 20n - 91 \in O(n^2)$, using C=8

Solve for k:

$$7n^2 + 20n - 91 \le 8n^2$$
$$20n - 91 \le n^2$$
$$n^2 - 20n + 91 \ge 0$$

Find roots: $n = \frac{(20 \pm (400 - 364)^{1/2})}{2}$

$= \frac{20 \pm 6}{2}$

$n = 13$ or $n = 7$

For all $n \ge 13$, the inequality holds.

Thus, the smallest value of K is **13**.

1. B) $7n^5 + 18n^4 - 3n^3 + 6n^2 - 2n + 5 = O(n^5)$

For polynomials:

Remove negative terms

$7n^5 + 18n^4 - 3n^3 + 6n^2 - 2n + 5 \le 7n^5 + 18n^4 + 6n^2 + 5$ (true for $n \ge 0$)

Raise to the highest power

$7n^5 + 18n^4 - 3n^3 + 6n^2 - 2n + 5 \le 7n^5 + 18n^5 + 6n^5 + 5n^5$ (true for $n \ge 1$)

$7n^5 + 18n^4 - 3n^3 + 6n^2 - 2n + 5 \le 36 n^5$

Thus, **C = 36 and K = 1**

2. **Solution**

2. Proof by induction (Limit of Logarithmic Functions)

Base Case: $n=0$:

$$\lim_{x\to\infty} \frac{(\ln x)^0}{x} = \lim_{x\to\infty} \frac{1}{x} = 0$$

which is true.

Inductive Step: Assume for $n=k$

$$\lim_{x\to\infty} \frac{(\ln x)^k}{x} = 0$$

Prove for $n=k+1$:

$$\lim_{x\to\infty} \frac{(\ln x)^{k+1}}{x} = \lim_{x\to\infty} \frac{\ln x \cdot (\ln x)^k}{x}$$

Apply L'Hopital's Rule:

Derivative of $(\ln x)^{k+1}$ is $(k+1)(\ln x)^k \cdot \frac{1}{x}$

Derivative of x is 1.

So,

$$\lim_{x\to\infty} \frac{(k+1)(\ln x)^k \cdot \frac{1}{x}}{1} = (k+1) \lim_{x\to\infty} \frac{(\ln x)^k}{x} = 0$$

Thus, by induction, the statement holds.

3. Pseudocode:

```
BINARYSEARCH( A[1, ……. , N], target) :
        low := 0
        high := n – 1
        while low ≤ high do:
                mid := Floor ( (low + high) / 2 )
                if A[mid] = target then:
                        return mid
                else if A[mid] < target then:
                        low := mid + 1
                else:
                        high := mid + 1
        return -1
```

4. For Algorithm 1

For loop a – iterates for n
        For loop b – iterates for n
                For loop c – iterates for n
                        For loop d – iterates for n
        For loop e – iterates for $n^4 – n^3$
                While loop f – iterates for (log n)

i) a: O(n), b: O(n), c: O(n), d: O(n), e: $O(n^4-n^3) = O(n^4)$, f: O(logn)

ii) Total runtime = a [(b*c*d) + (e*f)]
= O (n [(n*n*n*) + (n^4(logn)])
= O ($n^4 + n^5$ logn)

iii) Simplified expression = O (n^5(logn))

For Algorithm 2

For loop a – iterates for n

For loop b – iterates for log m

While loop c – iterates for m

While loop d – iterates for log m

Call subfunction – m^5 log m

For loop e – iterates for 3m

For loop f – iterates for $7n^2$

For loop g – iterates for constant time so 1

i) a: O(n), b: O(logm), c: O(m), d: O(logm), Function call: m^5logm, e: O(3m)= O(m), f: $O(7n^2)= O(n^2)$, g: 1

ii) Total runtime = a [(b*c*d) + Function call] + (e*f*g)

$= O\ (n\ [\ (\log m * m * \log m) + m^5 \log m\ ] + [m * n^2 * 1]\ )$

$= O\ (mn(\log m)^2 + m^5 n(\log m) + mn^2)$

iii) Simplified expression = $O\ (m^5 n(\log m)\ )$