CSE 321 - Homework 1

Bahadır Etk[redacted] Kılınç-1901042701

1)

$\underbrace{f(n) = 3(\log n+1)}_{T_1(n)}$

$\underbrace{g(n) = 4\log(\log n)}_{T_2(n)}$

$$\lim_{n\to\infty} \frac{3(\log n+1)}{4\log(\log n)} = \frac{\log n+1}{\log(\log n)} \xrightarrow{\text{LHopital}} \frac{\frac{1}{n.\ln 2}}{\frac{\frac{1}{n.\ln 2}}{\log n \ln 2}} = \log n = \infty$$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_1(n) > T_2(n)}$

---

2)

$\underbrace{f(n) = 4\log(\log n)}_{T_2(n)}$

$\underbrace{g(n) = n^5+8n^4}_{T_3(n)}$

$$\lim_{n\to\infty} \frac{4\log(\log n)}{n^5+8n^4} \Rightarrow \frac{\log(\log n)}{n^5} \xrightarrow{\text{LHospital}} \frac{\frac{1}{\log n.\ln 2}}{5n^4}$$

$$= \frac{1}{\log n.\ln 2.5n^4} = 0$$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ so $\boxed{T_3(n) > T_2(n)}$

$\underbrace{f(n) = n^5+8n^4}_{T_3(n)}$

$\underbrace{g(n) = 2000n+1}_{T_4(n)}$

$\lim_{n\to\infty} \frac{n^5+8n^4}{2000n+1}$ } Both are polynomial and power of up bigger than down $= \infty$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_3(n) > T_4(n)}$

$\underbrace{f(n) = \left(\frac{n}{6}\right)^2}_{T_5(n)}$

$\underbrace{g(n) = 2000n+1}_{T_4(n)}$

$\lim_{n\to\infty} \frac{n^2}{36(2000n+1)}$ } Both are polynomial and power of up bigger than down $= \infty$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_5(n) > T_4(n)}$

Bahadır Ekta Kılınç-1901042701 [redacted]

$\underbrace{f(n) = 2000n+1}_{T_4(n)}$

$\underbrace{g(n) = 4\log(\log n)}_{T_2(n)}$

$$\lim_{n\to\infty} \frac{2000n+1}{4\log(\log n)} \xrightarrow{\text{L'Hopital}} \frac{n'}{\log(\log n)'} = \frac{1}{\frac{1}{\log n \cdot \ln 2}} = \log n = \infty$$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_4(n) > T_2(n)}$

$\underbrace{f(n) = n^5 + 8n^4}_{T_3(n)}$

$\underbrace{g(n) = \left(\frac{n}{6}\right)^2}_{T_5(n)}$

$$\lim_{n\to\infty} \frac{n^5+8n^4}{\left(\frac{n}{6}\right)^2}$$ Both are polynomial and power of up bigger then down so $= \infty$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_3(n) > T_5(n)}$

$\underbrace{f(n) = 2^n + n^3}_{T_8(n)}$

$\underbrace{g(n) = n^5 + 8n^4}_{T_3(n)}$

$$\lim_{n\to\infty} \frac{2^n + \cancel{n^3}}{n^5 + \cancel{8n^4}} \Rightarrow \frac{2^n}{n^5} \xrightarrow{\text{L'Hopital}} \frac{2^n \cdot \ln 2}{5 \cdot n^4} \cdots\cdots \frac{(\ln 2)^5 \cdot 2^n}{L}$$

$= \infty$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_8(n) > T_3(n)}$

$\underbrace{f(n) = 3^n + n^2}_{T_6(n)}$

$\underbrace{g(n) = 2^n + n^3}_{T_8(n)}$

$$\lim_{n\to\infty} \frac{3^n + \underbrace{n^2}_{\text{ignored}}}{2^n + \underbrace{n^3}_{\text{ignored}}} \Rightarrow \frac{3^n}{2^n} = \infty$$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ so $\boxed{T_6(n) > T_8(n)}$

$\underbrace{f(n) = n^n + 1000n}_{T_7(n)}$

$\underbrace{g(n) = 3^n + n^2}_{T_6(n)}$

$$\lim_{n\to\infty} \frac{n^n + \underbrace{1000n}_{\text{ignored}}}{3^n + \underbrace{n^2}_{\text{ignored}}} = \frac{n^n}{3^n} = \infty$$

we find $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ so $\boxed{T_7(n) > T_6(n)}$

Result of all compare

$$\boxed{T_7 > T_6 > T_8 > T_3 > T_5 > T_4 > T_1 > T_2}$$

Bahadır Etka Kılınç - 1901042701
[redacted]

2

a) $f(n) = 99n$ and $g(n) = n$

$$L \longrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{99n}{n} = 99$$

we find L positive so $\underline{f(n) \in \Theta(g(n))}$.

b. $f(n) = 2n^4 + n^2$, $g(n) = (\log n)^6$

$$L \rightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{2n^4+n^2}{(\log n)^6} \xRightarrow{\text{L Hospital}} \frac{8n^3}{6 \cdot \frac{1}{n} \cdot \ln 2 \cdot (\log n)^5} \Rightarrow \frac{24n^4}{6(\log n)^5} \Rightarrow \frac{n^3}{5(\log n)^4 \cdot \frac{1}{n}}$$

$$\Rightarrow \frac{n^4}{(\log n)^4} \xrightarrow[\text{it goes}]{\text{-----}} \Rightarrow \frac{n^4}{1} = \infty$$ we find L is infinite so $\underline{f(n) \in \Omega(g(n))}$.

c. $f(n) = \sum_{x=1}^{n} x$, $g(n) = 4n + \log n$

$$L \rightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{n.(n+1)}{2.(4n+\log n)} = \frac{n^2 + \overset{\rightarrow \text{ignored}}{n}}{8n + \underset{\rightarrow \text{ignored}}{2\log n}} = \frac{n^2}{n} = \infty$$

we find L is infinite so $\underline{f(n) \in \Omega(g(n))}$

d. $f(n) = 3^n$, $g(n) = 5^{\sqrt{n}}$

$$L \rightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{3^n}{5^{\sqrt{n}}} = \frac{3^{n^{\frac{1}{2}}} \cdot 3^{n^{\frac{1}{2}}}}{5^{n^{\frac{1}{2}}}} = \left(\frac{3}{5}\right)^{n^{\frac{1}{2}}} \cdot 3^{n^{\frac{1}{2}}} = \left(\frac{9}{5}\right)^{n^{\frac{1}{2}}}$$

$$= \infty$$

we find L is infinite so $\underline{f(n) \in \Omega(g(n))}$

Bahadir Etka Kılınç -1901042701
[redacted]

3.

```
int myFunction (int nums [],int n) {
    for(int i=0; i<n; i++) {
        int count=1;
        for ( int j=i+1; j<n; j++)
            if (nums[j] == nums[i])
                count++;
        if (count > n/2)
            return nums[i];
    }
}
```

worst	Best
outer $\Rightarrow n$	outer for $\Rightarrow 1$
inner for $\Rightarrow n-1$	inner for $\Rightarrow n$
$1, 2, \dots n-1$	
$\frac{n.(n+1)}{2}$	$T(n) = n$
$T(n) = \frac{n.(n+1)}{2}$	
$T(n) \in \Theta(n^2)$	

a. "nums" parameter is integer array, "n" is number of element of nums array. Then returns the same value if half of the elements in the array have same value. Returns -1 if the values are not the same.

b. Worst Case: If no element occurs more than half the size of the array
Best Case: If first element occurs more than half of array's size.

4.

```
int myFunction2(int nums[], int n) {
    int i, *map, max=0;
    for (i=0 ; i<n ; i++) {
        if (nums[i] > max) {
            max = nums [i])
        }
    map = (int *) calloc (max+1, sizeof (int));
    for (i=0; i<n; i++)
        map [nums[i]]++;
    for (i=0; i<n;i++)
        if (map [nums[i]] > n/2)
            return nums[i];
    return -1;
}
```

worst	Best
$T(n) \in \Theta(n)$	$T(n) \in \Theta(n)$

a. Input "nums" is an array, "n" is size of "nums array. This algorithm allows us to find if there are more repeting elements in the array than the first half returns that repeating element otherwise returns -1.

b. Worst Case! If no element satisfies the condition.
Best Case: If first element satisfies the condition.

5.

Bahadır Erka Kılınç - 1901042701

[redacted]

	Question 3	Question 4
Worst Case	$\Theta(n^2)$	$\Theta(n)$
Best Case	$\Theta(n)$	$\Theta(n)$
Space Complexity.	No need extra space.	Extra space needed

In avarage time, the algoritm in question 4, works faster and more effectively then the algorithm in question 3. In other side, less memory is needed for question 3 if compared in terms of memory needed. As a result, although the algorithm in question 3 is less effective in terms of time complexity, it is more effective in terms of space complexity.

Bahadır Etka Kılınç - 1901042701
[redacted]

6.

a.
```
i=1, j=1, max = A_i x B_i
for i<=n do:
    for j <=m do:
        if max < A_i x B_j do:
            max = A_i x B_j
        j++
    i++
    j=1
```
Worst Case: $\Theta(n \times m)$
Best Case: $\Theta(n \times m)$

b.
```
i=1, j=1, array = [0 ..... n+m]
    for i <=n do;
        array = A;
        i++
    for j <=m do:
        array_{i+j} = B_j
        j++
    i = 1, j = 1
    for i <= n+m do:
        max = i
        j = i + 1
        for j <= n+m+1 do:
            if (array_i > array_max) do:
                max = j
        temp = array_max
        array_max = array
        array_i = temp
```
Worst Case: $\Theta((n+m)^2)$
Best Case: $\Theta((n+m)^2)$

Bahadır Etka Kılınç - 1901042701
[redacted]

c.

A_{n+1} = new_element

Worst Case : $\Theta(1)$

Best Case : $\Theta(1)$

d. $i = 1$

for $i <= n$ do:

  if (A_i == element) do:

    $j = i$

  for $j < n$ do:

    $A_j = A_{j+1}$

  A_j = null

  break

Worst Case : $\Theta(n)$

Best Case : $\Theta(n)$