Q1. Solution:

$$\underline{T_2} < \underline{T_1} < \underline{T_4} < \underline{T_5} < \underline{T_3} < \underline{T_8} < \underline{T_6} < T_7$$

Proofs:

1. $$\lim_{n\to\infty} \frac{T_2}{T_1} = \frac{4\log(\log n)}{3\log n + 3}$$

$$= 4 \cdot \lim_{n\to\infty} \frac{\ln(\ln(n))}{3\ln n + 3}$$

$$= 4 \cdot \lim_{n\to\infty} \frac{\frac{\ln(\ln(n))}{\ln(n)}}{3 + \frac{3}{\ln(n)}}, \quad \frac{\lim_{n\to\infty} \frac{\ln(\ln(n))}{\ln(n)}}{\lim_{n\to\infty} 3 + \frac{3}{\ln(n)}}$$

$$\lim_{n\to\infty} \frac{\ln(\ln(n))}{\ln(n)} = \frac{\frac{\frac{1}{n}}{\ln(n)} \cdot \frac{1}{\cancel{n}}}{\frac{1}{\cancel{n}}} = \frac{\ln(n)}{n} \quad \text{(l'hospital)}$$

$$\Rightarrow \frac{(\ln(n))'}{n'} = \frac{\frac{1}{n}}{1} = \frac{1}{n} \qquad \lim_{n\to\infty} \frac{1}{n} = 0$$

$$T_2 = O(T_1(n))$$

② $\lim_{n\to\infty} \frac{T1}{T4} = \lim_{n\to\infty} \frac{3\log n+3}{2000n+1}$ (divide by n)

$$= \frac{3\ln+3}{n} \Big/ 2000 + 1/n \to \frac{\lim_{n\to\infty} \frac{3\ln+3}{n}}{\lim_{n\to\infty} 2000+\frac{1}{n}} = 0_{//}$$

$$\lim_{n\to\infty} \frac{3\ln+3}{n} \longrightarrow \frac{\cancel{\frac{3}{n}\cdot n} - 3\ln - \cancel{3}}{n^2} = \frac{-3\ln}{n^2}$$

$$\left(\frac{-3\ln}{n^2}\right)' = \frac{-\frac{3}{n}}{2n} = \frac{-3}{2n^2} \quad \lim_{n\to\infty} \frac{-3}{2n^2} = 0$$

$$\lim_{n\to\infty} 2000+\frac{1}{n} = \underbrace{\lim_{n\to\infty} 2000}_{2000} + \underbrace{\lim_{n\to\infty} \frac{1}{n}}_{0} = 2000$$

$T_1(n) = O(T_4(n))$

③ $\lim_{n\to\infty} \frac{T4}{T5} = \lim_{n\to\infty} \frac{2000n+1}{\left(\frac{n}{6}\right)^2}$

$$= \frac{(2000n+1)6^2}{n^2} \to 36 \cdot \lim_{n\to\infty} \frac{2000n+1}{n^2}$$

$$\to 36 \cdot \left(\underbrace{\lim_{n\to\infty} \frac{2000}{n}}_{0} + \underbrace{\lim_{n\to\infty} \frac{1}{n^2}}_{0} \right) = 0$$

$T_4(n) = O(T_5(n))$

④ $\lim_{n\to\infty} \frac{T5}{T3} = \left(\frac{n}{6}\right)^2 \Big/ n^5 + 8n^4$

$$\frac{n^2}{36(n^5+8n^4)} \to \frac{\frac{n^2}{n^2}}{\frac{36(n^5+8n^4)}{n^2}} = \frac{1}{36(n^3+8n^2)}$$

$$\to \frac{1}{36} \cdot \lim_{n\to\infty} \frac{1}{n^3+8n^2} = 0$$

$T5(n) = O(T3(0))$

⑤ $\lim_{n\to\infty} \frac{T3}{T8} = \frac{n^5+8n^4}{2^n+n^3}$ (divide by n^3)

$$\frac{\frac{n^5+8n^4}{2^n}}{1+\frac{n^3}{2^n}} \qquad \frac{\lim_{n\to\infty} \frac{n^5+8n^4}{2^n}}{\lim_{n\to\infty} 1+\frac{n^3}{2^n}} = 0$$

$\to 0$

$\swarrow$ 0 değil.

$\downarrow$ 1

$$\frac{5n^4+32n^3}{2^n \cdot \ln 2} \quad \frac{20n^3+96n^2}{2^n \cdot \ln^2 2} \quad \frac{60n^2+192n}{2^n \cdot \ln^3 2} \quad \frac{120n+192}{2^n \cdot \ln^4 2} \to \frac{120}{2^n \cdot \ln^5 2} \to 0$$

$$\lim_{n\to\infty} 1 + \lim_{n\to\infty} \frac{n^3}{2^n} \quad \frac{3n^2}{2^n \ln 2} \quad \frac{6n}{2^n \ln^2 2} \to \frac{6}{2^n \ln^3 2} = 0 \qquad = 1$$

$T_3(n) = T_8(O(n))$

6 $\lim_{n\to\infty} \frac{T_8}{T_6} = \lim_{n\to\infty} \frac{2^n + n^3}{3^n + n^2}$

$$\lim_{n\to\infty} \frac{\left(\frac{2}{3}\right)^n + \frac{n^3}{3^n}}{\underbrace{1 + \frac{n^2}{3^n}}_{\to 0 \text{ değil}}} \longrightarrow \frac{\overbrace{\lim_{n\to\infty} \left(\frac{2}{3}\right)^n + \frac{n^3}{3^n}}^{0}}{\lim_{n\to\infty} 1 + \frac{n^2}{3^n}}$$

$$\lim_{n\to\infty} \left(\frac{2}{3}\right)^n = 0 \qquad \lim_{n\to\infty} \frac{n^3}{3^n} = 0$$

$$\underbrace{\lim_{n\to\infty} 1}_{1} + \underbrace{\lim_{n\to\infty} \frac{n^2}{3^n}}_{0} \longrightarrow 1 \qquad \frac{0}{1} = 0$$

$T_8(n) = O(T_6(n))$

7 $\lim_{n\to\infty} \frac{T_6}{T_7} = \lim_{n\to\infty} \frac{3^n + n^2}{n^n + 1000n}$ (divide by n^n)

$$\frac{\frac{3^n}{n^n} + n^{2-n}}{\underbrace{1 + 1000n^{1-n}}_{0 \text{ değil}}} \rightarrow \frac{\lim_{n\to\infty} \left(\frac{3^n}{n^n} + n^{2-n}\right) \longrightarrow 0}{\lim_{n\to\infty} \left(1 + 1000n^{1-n}\right) \longrightarrow 1} = 0 \,//$$

$T_6(n) = O(T_7(n))$

To sum;

$$T_2 < T_1 < T_4 < T_5 < T_3 < T_8 < T_6 < T_7$$

Q2. Solution :

a) $f(n) = 99n$

$g(n) = n$

$\lim_{n \to \infty} \frac{99\cancel{n}}{\cancel{n}} \to 99 \ (c>0)$ $\quad f(n) \in \Theta(g(n))$

b) $f(n) = 2n^4 + n^2$

$g(n) = (\log n)^6$

$\lim_{n \to \infty} \frac{2n^4 + n^2}{(\log n)^6} \to \frac{\infty}{\infty}$ (L'hospital) need multiple derivations.

$\frac{8n^3 + 2n}{\frac{6 \ln^5}{n \cdot \ln}} \to \frac{(8n^3+2n)n \cdot \ln}{6 \cdot \ln^5} \quad \frac{(8n^4+2n^3)\cancel{\ln}}{6 \cancel{\ln}^5}$

taking derivations till the logn goes to constant.

numerator still has n. which goes to ∞.

$\frac{\infty}{c} = \infty$

$f(n) \in \Omega(g(n))$

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

$$\lim_{n\to\infty} \frac{(1+2+3+..+n)}{4n+\log n} \to \frac{\frac{n.(n+1)}{2}}{4n+\log n}$$

$$\frac{n^2+n}{8n+2\log n.} \to \frac{n+1}{\underbrace{8+2\log n}_{\text{not } 0.}} \quad (\text{L'hopital})$$

$$\frac{1}{2/n.\ln 2} \to \frac{n.\ln 2}{2} \to \frac{\infty}{c} = \infty$$

$f(n) \in \Omega(g(n))$

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

$$\lim_{n\to\infty} \frac{3^n}{5^{\sqrt{n}}} \to \frac{3^n}{5^{1/2n}} \to 3^n . 5^{-1/2n}$$

$$3^n . \ln 3 . 5^{-1/2n} + -1/2 . 5^{-1/2n} . \ln 5 . 3^n$$

$$3^n . 5^{-1/2n} (\ln 3 + -1/2 . \ln 5) \to \infty$$

$f(n) = \Omega(g(n))$

Q3. Solution:

a) This algorithm takes an array and an index number as **inputs**, gives if the array includes repeated numbers from start of the array till the given index at more than half of index times.

$\{\underbrace{1, 2, 3, 2,}\ 4, 5, 3\}$, $n = 4.$ (an example)

2 recurrence of 2. , ? > 4/2 , no. returns -1.] output

$\{\underbrace{3, 3, 3, 2, 1,}\ 3, 5, 6\}$, $n = 5$ (an example)

3 recurrence of 3. , ? > 5/2 , yes. returns the number (3).] output.

b)

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

(n-i) times

n times

Total number of nested loop operation $= (n-1)+(n-2)+(n-3)+2+1$

$=$ sum of arithmetic series $\rightarrow n.(n-1)/2 = n^2/2 - n/2 \rightarrow O(n^2)$

At each step of iteration, loops are doing $O(1)$ operations.

So overall time complexity of this algorithm $= O(n^2) * O(1) = O(n^2)$

— Worst & Best Cases for this algorithm —

- For worst case analysis, it is enough to calculate Big-O notation to get upperbound for algorithm. It is $O(n^2)$.

- For best case analysis, lowerbound must be analyzed. On this algorithm, there are some cases which will make this algorithm on best case.

  - if $n = 1$,

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

1 time

doesn't loop.

→ Outer loop executes 1 time. inner loop can not be executed because of condition, so it is $\Omega(1)$.

- n can be < 1. at this point, also outer for loop is not executed. Just return statement works, still $\Omega(1)$.

Çağla Şahin
171044050
HW1 Solutions