Ahmet Furkan Kurban
1801042674
Ahmet

1) a) $(2^n+n^3) \in O(4^n)$

$O(g(n)) = \{f(n) \mid \exists c>0, n_0 \ni \forall n \geq n_0, 0 \leq f(n) \leq c \cdot g(n)\}$

$2^n+n^3 \leq c * 4^n$ for $c=1$ c is a positive constant

$2^n+n^3 \leq 4^n$ $n_0=0 \Rightarrow n \geq 0$ and no is a non negative integer.

we find an c and no value.

b) $\sqrt{10n^2+7n+3} \in \Omega(n)$

$f(n) \geq c * g(n)$ for all $n \geq n_0$ c is a positive constant and n_0 is a non-negative integer.

$\sqrt{10n^2+7n+3} \geq c * n$ For $c=1$ find an c and n_0

$n_0=0 \Rightarrow n \geq 0$ value that ensure equality.

c) $n^2+n \in o(n^2)$ $f(n)=o(g(n)) \Leftrightarrow \forall c>0, \exists n_0 \ni \forall n \geq n_0,$

$f(n) \leq c \cdot g(n)$

$n^2+n < c * n^2$ For $c=3$

$n^2+n < 3n^2$ $n_0=1 \Rightarrow n \geq 1$

CamScanner ile tarandı

d) $3\log_2^2 n \in \Theta(\log_2 n^2)$

$t(n) = \Theta(g(n))$

$c2 * g(n) \leq t(n) \leq c1 * g(n)$ for all $n \geq n_0$

c_1 and c_2 are positive constants and n_0 is a non-negative integer

$c2 * \log_2 n^2 \leq 3\log_2^2 n \leq c_1 * \log_2 n^2$

$c2 * 2 \cdot \cancel{\log_2 n} \leq 3 \cdot (\cancel{\log_2 n} * \log_2 n) \leq c_1 * 2 \cdot \cancel{\log_2 n}$

$c2 * 2 \leq 3 \cdot \log_2 n \leq c_1 * 2$

$\frac{c2*2}{3} \leq \log_2 n \leq \frac{c_1 * 2}{3}$ $\quad c_1 = 6$, $c_2 = 3$ $\Rightarrow 2 \leq \log_2 n \leq 4$

$2^2 \leq n \leq 4^2$

$4 \leq n \leq 16$

$n \geq n_0$ is not true for any c and n_0 because $\underline{\log_2^2 n > \log_2 n^2}$

There is no such n_0 value that ensure $n \geq n_0$

e) $(n^3+1)^6 \in O(n^3)$

$n^{18}+1 \in O(n^3)$

$t(n) \leq c * n^3 \quad n \geq n_0$

$n^{18}+1 \leq c * n^3$

There is no such a c value to ensure $n^{18}+1 \leq n^3$ because $n^{18}+1 > n^3$ //

2)

a) $2n\log(n+2)^2 + (n+2)^2 \log\frac{n}{2}$

$2.2.n\log(n+2) + (n+2)^2\log\frac{n}{2} = c * g(n)$ For $\theta(g(n))$ so we can find by finding $O(n)$ and $\Omega(n)$ value.

the biggest complexity value is $n^2\log(\frac{n}{2})$

then if we interrupt constant value we can find

$O(n^2.\log n)$ so $n_0 \to \forall n \geq n_0$

$n^2\log n \leq c_1 * g(n)$ for $c_1 = 1$

$n^2\log n \leq g(n)$ so $g(n) = O(n^2\log n)$

$\Omega(n^2\log n)$ so $n_0 \to \forall n \geq n_0$

$n^2\log n \geq c_1 * g(n)$ for $c_1 = 1$

$n^2\log n \geq g(n)$

so we can say $g(n) = \Omega(n^2\log n)$

if we connect then simplest $g(n)$ can be $\theta(n^2\log n)$ //

CamScanner ile tarandı

b) $0{,}001n^4 + 3n^3 + 1$

For $\Theta(g(n))$ value firstly we have to find $O(n)$ and $\Omega(n)$ for $g(n)$ then we can combined them.

$0{,}001n^4$ is the highest complexity value and if we ignore constant value we can find n^4 then. For $O(n)$

$n^4 \leq c_1 \cdot g(n)$ $c_1 = 1$

$n^4 \leq g(n)$ so

$g(n)$ can be $O(n^4)$

$n^4 \geq c_1 \cdot g(n)$ $c_1 = 1$

$n^4 \geq g(n)$ so $g(n)$ can be $\Omega(n^4)$ so

if we connect them simplest $g(n)$ value can be $\Theta(n^4)$

3) a) $\log n$, $n^{\log n}$, $n^{1.5}$

$$\lim_{n\to\infty} \frac{\log n}{n^{1.5}} = \frac{\infty}{\infty} \Rightarrow \lim_{n\to\infty} \frac{\frac{1}{n}}{1.5n^{0.5}} = \lim_{n\to\infty} \frac{1}{1.5n^{1.5}} = 0 \Rightarrow n^{1.5} > \log n$$

$$\lim_{n\to\infty} \frac{n^{\log n}}{n^{1.5}} = \frac{\infty}{\infty} \Rightarrow \lim_{n\to\infty} n^{\log n - 1.5} = \infty \Rightarrow n^{\log n} > n^{1.5}$$

so $n^{\log n} > n^{1.5} > \log n$ //

b) 2^n, $n!$, n^2

$$\lim_{n\to\infty} \frac{n^2}{2^n} = \frac{\infty}{\infty} \Rightarrow \lim_{n\to\infty} \frac{2n}{2^n . \ln 2} = \frac{\infty}{\infty} \Rightarrow \lim_{n\to\infty} \frac{2}{2^n . \ln 2 . \ln 2} = 0 \Rightarrow 2^n > n^2$$

$$\lim_{n\to\infty} \frac{n!}{2^n} = \lim_{n\to\infty} \frac{\sqrt{2\pi n} . \left(\frac{n}{e}\right)^n}{2^n} = \lim_{n\to\infty} \sqrt{2\pi n} \cdot \left(\frac{n}{2e}\right)^n = \infty$$

$\Rightarrow n! > 2^n$

so $n! > 2^n > n^2$

c) $n\log n$, $\sqrt{n}$

$$\lim_{n\to\infty} \frac{n\log n}{\sqrt{n}} = \frac{\infty}{\infty} = \lim_{n\to\infty} \frac{\log n + \cancel{n}\frac{1}{\cancel{n}}}{\frac{1}{2}n^{-\frac{1}{2}}} = \lim_{n\to\infty} \frac{2(\log n + 1)\sqrt{n}}{1} = \infty$$

$n\log n > \sqrt{n}$

d) $n.2^n$, 3^n

$$\lim_{n\to\infty} \frac{n.2^n}{3^n} = \frac{\infty}{\infty} \Rightarrow \lim_{n\to\infty} n\left(\frac{2}{3}\right)^n = \left(\frac{2}{3}\right)^n + n\left(\frac{2}{3}\right)^n . \ln\left(\frac{2}{3}\right)$$

$$\lim_{n\to\infty} \left(\frac{2}{3}\right)^n \left(1 + \ln\frac{2}{3}\right) = \infty$$

$n.2^n > 3^n$

e) $\sqrt{n+10}\ ,\ n^3$

$$\lim_{n\to\infty} \frac{\sqrt{n+10}}{n^3} = \frac{\infty}{\infty} = \lim_{n\to\infty} \frac{1}{2\sqrt{n+10}\cdot 3n^2} = 0 \Rightarrow n^3 > \sqrt{n+10}$$

4)

a) The basic operation is the comparison in the inner loop.

$$\Downarrow$$

$B[i,j] \mathrel{!}= B[j,i]$

constant

b)
$$C_{worst}(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1)-(i+1)+1] = \sum_{i=0}^{n-2} (n-1-i)$$

$$= \sum_{i=0}^{n-2} n-1 - \sum_{i=0}^{n-2} i = (n-1)\boxed{\sum_{i=0}^{n-2} 1} - \frac{(n-2)(n-1)}{2}$$

$$= (n-1)^2 - \frac{(n-2)(n-1)}{2} = \frac{(n-1)n}{2} = \frac{1}{2}n^2$$

c) $\lim_{n\to\infty} \frac{1}{n^2/2} = \lim_{n\to\infty} \frac{2}{n^2} = 0$ so Complexity is $O(n^2)$

5) a) The basic operation is the multiplication operation in the innermost loop.

$A[i,k] * B[k,j]$

b) $$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} 1 = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} n = \sum_{i=0}^{n-1} n^2 = n^3$$

c) Time complexity is $O(n^3)$

6)
```
Void Checkfair (int A[], int size, int target){
    For( int i=0 ; i<size-1; i++){ → T1(n)
        For(int j=i+1; j<size; j++){ → T2(n)
            if(A[i] + A[j] == target){ Θ(1)
                cout<<"("<<A[i]<<","<<A[j]<<")"<<endl; Θ(1)
            }
        }
    }
}
```

$$T(n) = \underset{\Theta(n)}{\underset{\downarrow}{T_1(n)}} * \underset{\Theta(n)}{\underset{\downarrow}{T_2(n)}} = T(n) = \Theta(n^2)$$