1-) a) $(2^n+n^3) \in O(4^n) \Rightarrow \boxed{\text{True}}$ ✓

iff there exist positive constant c and n_0 st. $f(n) \le c.g(n)$ whenever $n \ge n_0$

$f(n) = 2^n+n^3$ assume $c=13$, $n_0=4$

$g(n) = 4^n$

$64 \cdot 4 = 256$

$2^4+4^3 \le 4^4 \Rightarrow 16+64 \le 256$

$\boxed{80 \le 256}$ ✓

---

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

iff there exist positive constant c and n_0 st. $c.g(n) \le f(n)$ $n \ge n_0$

$f(n) = \sqrt{10n^2+7n+3}$

$g(n) = n$

a.t. $\boxed{c=3,\ n_0=6}$

$3.6 \le \sqrt{360+42+3}$

$18 \le \sqrt{405}$

$\boxed{\sqrt{324} \le \sqrt{405}}$ ✓

$18 \times 18 = 144 + 18\_ = 324$

---

c) $n^2+n \in o(n^2) = \boxed{\text{False}}$

iff there exist positive integer c and n_0 s.t. $f(n) < c.g(n)$ whenever $n \ge n_0$

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

$g(n) = n^2$

a.t

$c = a,\ a>0$

$n_0 = n_0,\ n_0>0$

$n_0^2+n_0 < a.n_0^2$

$n_0 < (1-a)$ ✗

there is no positive 'a' number to make this equation true

---

d) $3\log_2^2 n \in \Theta(\log_2 n^2) \Rightarrow \boxed{\text{False}}$

iff there exist positive constant c_1, c_2 & n_0 st. $c_1.g(n) \le f(n) \le c_2.g(n)$ wheever $n \ge n_0$

$f(n) = 3\log_2^2 n = 3\log_2(\log_2 n)$

$g(n) = \log_2 n^2$

a.t.

$c_1 = a;\ a>0$

$c_2 = b,\ b>0$

$n_0 = n_0,\ n_0>0$

$a.\log_2 n^2 \le 3.\log_2(\log_2 n) \le b.\log_2 n^2$

$\log_2 n^2 \le \log_2(\log_2 n) \le \log_2 n^2$ eliminate constants

✗ There is no any positive number for 'a' & 'b' to make this equation True

e) $(n^3+1)^6 \in O(n^3)$ = $\boxed{\text{False}}$ $f(n) \leq c \cdot g(n)$, $n \geq n_0$

$f(n) = (n^3+1)^6$
$g(n) = n^3$

$\lim_{n \to \infty} \frac{(n^3+1)^6}{n^3} = \infty$ so $(n^3+1)^6 > n^3$

- there is no any positive integers for c and n_0

---

2)

a) $2n \log (n+2)^2 + (n+2)^2 \log \frac{n}{2} = f(n)$ $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot f(n)$

$n \log n^2 + (n^2+4n) . \log n \Rightarrow$ eliminate constants
$\times 2, +2, +4, \frac{1}{2}$

$\Rightarrow n \log n + n^2 \log n + 4n \log n$

eliminate low orders

$\Rightarrow \boxed{n^2 \log n} \in \Theta(n^2 \log n)$

so

$$\boxed{2n \log n (n+2)^2 + (n+2)^2 \log \frac{n}{2} \in \Theta(n^2 \log n)}$$

When we want to simplify an equation. We need to eliminate constant and low orders.

---

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

$n^4 + \cancel{n^3} \Rightarrow$ eliminate constants, and low orders
$0{,}001, 3, +1$ n^3

$\Rightarrow n^4 \in \Theta(n^4)$

so

$0{,}0001 n^4 + 3n^3 + 1 \in \Theta(n^4)$

Scanned with CamScanner

3) a) $\log n$, $n^{\log n}$, $n^{1,5}$

$y = \log_a g(x) \Rightarrow y' = \dfrac{g'(x)}{g(x) \cdot \ln a}$

$$\lim_{n\to\infty} \frac{n^{\log n}}{n^{1,5}} \Rightarrow \lim_{n\to\infty} n^{(\log n - 1,5)} \Rightarrow \infty \quad \text{so}_{n\to\infty} \quad \boxed{n^{\log n} > n^{1,5}}$$

$$\lim_{n\to\infty} \frac{n^{1,5}}{\log n} = \frac{\frac{3}{2}\cdot\sqrt{n}}{\frac{1}{n \ln(n)}} = \frac{3}{2}\cdot\sqrt{n}\cdot n \ln(n) \quad \text{so}_{n\to\infty} \quad \boxed{n^{1,5} > \log n}$$

$$\boxed{n^{\log n} > n^{1,5} > \log n}$$

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

$n! \cong \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n$

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

$\text{so}_{n\to\infty}$ $\boxed{n! > 2^n}$

$y = a^{g(x)}$

$y' = g'(x) \cdot a^{g(x)} \cdot \ln a$

$y = f(x) = h(x) \cdot g(x)$

$y' = h'(x) \cdot g(x) + h(x) \cdot g'(x)$

$$\lim_{n\to\infty} \frac{2^n}{n^2} \Rightarrow \frac{1 \cdot 2^n \cdot \ln 2}{2n} \Rightarrow$$

$$\Rightarrow \frac{\ln 2 \cdot 2^n \cdot \ln 2 + \cancel{0 \cdot 2^n}}{\underbrace{2 \cdot 1}_{\text{goes constan}}} \Rightarrow$$

goes constan so $n \to \infty$ $\Rightarrow \infty$

$\boxed{2^n > n^2}$

$$\boxed{n! > 2^n > n^2}$$

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

$$\lim_{n\to\infty} \frac{(n\log n)'}{(\sqrt{n})'} = \frac{1 \cdot \log n + \frac{1}{\cancel{x} \cdot \ln(10)} \cdot \cancel{x}}{\frac{1}{2\sqrt{x}}} \Rightarrow \log n + \frac{1}{\ln(10)} + 2\sqrt{x}$$

$\Rightarrow \infty$ so that

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

d) $n2^n, 3^n$

$$\lim_{n\to\infty} \frac{(n2^n)'}{(3^n)'} \Rightarrow \lim_{n\to\infty} n \cdot \left(\frac{2}{3}\right)^n \Rightarrow \lim_{n\to\infty} \frac{x}{\frac{1}{\left(\frac{2}{3}\right)^x}} \xRightarrow{L'}$$

$$\Rightarrow \lim_{n\to\infty} \frac{1}{\underbrace{\left(\frac{3}{2}\right)^x \ln\left(\frac{3}{2}\right)}_{\to\infty}} = 0$$

$$\boxed{n2^n = 3^n}$$

e)

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

$$\lim_{n\to\infty} \frac{\sqrt{n+10}}{n^3} \Rightarrow \frac{(n+10)^{\frac{1}{2}}}{n^3} \Rightarrow \frac{\frac{1}{2} \cdot \frac{1}{\sqrt{n+10}}}{n^3} \Rightarrow \lim_{n\to\infty} \frac{\frac{1}{2\sqrt{n+10}}}{n^3}$$

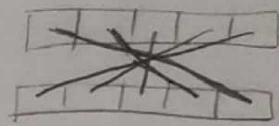
$$\Rightarrow \lim_{n\to\infty} \frac{1}{2\sqrt{n+10} \cdot n^3} = 0$$

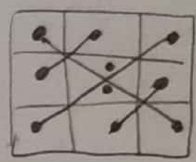
$$\boxed{n^3 > \sqrt{n+10}}$$

Scanned with CamScanner

4) algoritm1(B[0...,n-1, 0 .....n-1])

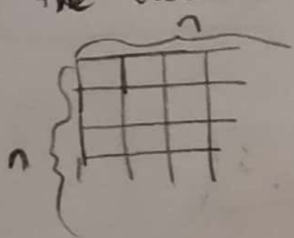
```
for i=0 to n-2 do
    for j=i+1 to n-1 do
        if B[i,j] != B[j,i]
            return false
return True
```





- The algoritm checking for the given array is symmetric matrix or not.

- The worst case when occurs if the given matrix is symmetric matrix. if it is two for loop iterate all over the two dimensional array.



$$\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} 1 \Rightarrow$$

Note: $\sum_{i=m}^{n} 1 = n-m+1$

$$\Rightarrow (n-1)-(i+1)+1$$

$$\Rightarrow \sum_{i=0}^{n-2}((n-1)-(i+1)+1) \Rightarrow \sum_{i=0}^{n-2} n-1-i$$

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

$$\Rightarrow n^2 \cancel{-n} - \frac{n^2-3n+3}{2} \cancel{+n} + 2 \Rightarrow n^2 - \frac{n^2}{2} - \frac{3n}{2} + \frac{3}{2} + 2 \Rightarrow \boxed{\frac{n^2-3n+7}{2}}$$

$$\boxed{W(n) = \Theta(n^2)} \qquad \frac{n^2-3n+7}{2} \in \Theta(n^2)$$

a) It is checking the given array is symmetric or not.
Basic operation is comparision.

b) $n^2 - \frac{n^2}{2} - \frac{3n}{2} + \frac{3}{2} + 2$ times basic operation executed

c) $T(n) = O(n^2)$

5) algoritm2 (A [0... n-1, 0... (n-1)] , B [0... n-1, 0... n-1])

```
for i=0 to n-1 do
    for j=0 to n-1 do
        C[i,j] = 0.0
        for k=0 to n-1 do
            C[i,j] = C[i,j] + A[i,k] * B[k,j]
return C
```

a) Its basic operation is assignment (=)

b) for first = operator

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} 1 \Rightarrow \sum_{i=0}^{n-1} n \Rightarrow n.n \Rightarrow \boxed{n^2}$$

Note:

$$\sum_{i=m}^{n} 1 = n-m+1$$

for second = operator

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

Total basic operation executed time = $\boxed{n^3+n^2} \in O(n^3)$

c) As we said in(b) there are 3 nested for loop from 0 to n-1

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} \Rightarrow n^3$$

$$\boxed{T(n) = \Theta(n^3)}$$

Scanned with CamScanner

6)

```
algoritm 3 ( List [0 .... n-1], number)
        pairs = []
        for i=0 to n-1 do
            for j= i+1 to n-1 do
                if L[i] * L[j] = number
                    pairs. add ((L[i], L[j]))
            endfor
        endfor
        return pairs;
end of function
```

Time complexity:

Note:

$$\sum_{i=m}^{n} 1 = n-m+1$$

$$\sum_{i=0}^{n-1} \underbrace{\sum_{j=i+1}^{n-1} 1}$$

$$\sum_{i=0}^{n-1} ((n-1)-(i+1)+1) \Rightarrow \sum_{i=0}^{n-1} n-1-i \Rightarrow$$

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

$$\Rightarrow n^2 - \frac{n^2-n}{2} - n + 1 \Rightarrow \boxed{\frac{n^2-3n+2}{2}} \in \Theta(n^2)$$

Muhammed Bedir

ULUCAY 1801042697