1.

a) $2^n + 3 \overset{?}{\in} O(4^n)$ n_0 is a positive integer, c is a positive constant

$0 \leq 2^n + 3 \leq c4^n$ for $\forall n \geq n_0$ $\boxed{n_0 = 1}$

$\frac{2^n+3}{4^n} \leq c$ for $\forall n \geq 1$ $\boxed{c = 2}$

$2^n + 3 \leq 2.4^n$, $1 + \frac{3}{2^n} \leq 2^{n+1}$, $1 \leq \frac{2^{2n+1} - 3}{2^n}$ for $\forall n \geq 1$

$2^n + 3 \in O(4^n)$ is TRUE

b) $\sqrt{10n^2+7n+3} \overset{?}{\in} \Omega(n)$ n_0 is a positive integer, c is a positive constant

$0 \leq cn \leq \sqrt{10n^2+7n+3}$ for $\forall n \geq n_0$ $\boxed{n_0 = 1}$

$c^2n^2 \leq 10n^2+7n+3$, $c^2 \leq 10 + \frac{7}{n} + \frac{3}{n^2}$, $c \leq \sqrt{10 + \frac{7}{n} + \frac{3}{n^2}}$ for $\forall n \geq 1$

$\boxed{c = 3}$

$3n \leq \sqrt{10n^2+7n+3}$ for $\forall n \geq 1$

$9n^2 \leq 10n^2+7n+3$, $0 \leq n^2+7n+3$ for $\forall n \geq 1$

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

c) $n^2+n \overset{?}{\in} o(n^2)$ n_0 is a positive integer.
c is a positive constant.

For every positive constant c, there is a positive integer n_0 such that $n^2+n < cn^2$ ①

For $c=1$:

$n^2+n < n^2$, $n<0$

$n^2-n^2+n<0$

If I assume $c=1$, there is no positive integer n_0 possible. Then, $n^2+n \in o(n^2)$ is false. Because, ① statement is not satisfied.

$n^2+n \in o(n^2)$ is FALSE

d) $3\log_2^2 n \overset{?}{\in} \Theta(\log_2 n^2)$ n_0 is a positive integer.
c_1 and c_2 are positive constants.

$\log_2 n^2 = 2\log_2 n$, $\log_2^2 n = \log \log_2 n$

$$0 \leq \underbrace{c_1 2\log_2 n \leq \overbrace{3\log_2^2 n}^{}}_{\Omega} \overbrace{\leq c_2 2\log_2 n}^{O}$$ for $\forall n \geq n_0$

Ω part:

$2c_1 \log_2 n \leq 3\log_2^2 n$, $\frac{2c_1\log_2 n}{2\log_2 n} \leq \frac{3\log_2^2 n}{2\log_2 n}$, $\underbrace{c_1}_{LHS} \leq \underbrace{\frac{3}{2} \cdot \frac{\log_2^2 n}{2\log_2 n}}_{RHS}$ ①

Let $n \to \infty$, In ① RHS keeps decreasing to 0. After n passes n_0 LHS $\geq$ RHS. So, ① can't be true. Also, from formal definition "The values of c and n_0 must be fixed and must not depend on n." For ① I can't find a fixed c. Since, RHS keeps decreasing to 0.

For $n>1$
$2\log_2 n > \log_2^2 n$
growth rate of $2\log_2 n$ is bigger than $\log_2^2 n$

$3\log_2^2 n \in \Theta(\log_2 n^2)$ is FALSE

e) $(n^3+1)^6 \overset{?}{\in} O(n^3)$ n_0 is a positive integer
c is a positive constant

$0 \leq (n^3+1)^6 \leq n^3 . c$ for $\forall n \geq n_0$

$$\frac{(n^3+1)^6}{n^3} \leq c \quad (1)$$

Let $n \to \infty$. In (1) $LHS = \frac{(n^3+1)^6}{n^3}$ and $RHS = c$.

In (1) LHS keeps increasing to infinity. But, RHS is constant. After n passes n_0 LHS $\geq$ RHS. So, (1) can't be true. With that, $(n^3+1)^6 \in O(n^3)$ is FALSE.

Also, from formal definition: "The values of c and n_0 must be fixed and must not depend on n." As you can see, LHS keeps increasing to infinity and RHS is constant. So, i can't find a fixed c for $\forall n \geq n_0$.

Scanned with CamScanner

2.

a) $f(n) = 2n\log(n+2)^2 + (n+2)^2\log\frac{n}{2} = 4n\log(n+2) + (n+2)^2\log\frac{n}{2}$

$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for $\forall n \geq n_0$

$f(n) \leq c_2 g(n)$, $\frac{f(n)}{g(n)} \leq c_2$ ① $LHS = \frac{f(n)}{g(n)}$

$RHS = c_2$

In ① let $n \to \infty$, we need to find a fixed c_2. That is possible only if LHS is constant as $n \to \infty$. Growth rate of $f(n)$ and $g(n)$ have to be equal. $\max(g(n)) = \max(f(n))$ has to be true. Simplest $g(n)$ would be $\max(f(n))$.

$\max(f(n)) = \max(2n\log(n+2)^2 + (n+2)^2\log\frac{n}{2}) = (n+2)^2\log\frac{n}{2} \in O(n^2\log n)$

So, $g(n) \in O(n^2\log n)$ and simplest $g(n) = n^2\log n$.

b) $f(n) = 0{,}001n^4 + 3n^3 + 1$, $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for $\forall n \geq n_0$

$f(n) \leq c_2 g(n)$, $\frac{f(n)}{g(n)} \leq c_2$ ① $LHS = \frac{f(n)}{g(n)}$

$RHS = c_2$

I will solve this like i did on part a.

$\max(f(n)) = \max(0{,}001 . n^4 + 3n^3 + 1) = 0{,}001n^4 \in O(n^4)$

So, $g(n) \in O(n^4)$ and simplest $g(n) = n^4$

3.

a) $\lim_{n \to \infty} \frac{n^{1,5}}{n^{\log n}} = 0$, $\lim_{n \to \infty} \frac{\log n}{n^{1,5}} = 0$

$n^{\log n} > n^{1,5} > \log n$ $n^{1,5} \in O(n^{\log n})$ and $\log n \in O(n^{1,5})$

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

$2^n \in O(n!)$ $\lim_{n \to \infty} \frac{2^n}{n^2} \underset{\text{L'Hospital}}{=} \lim_{n \to \infty} \frac{\ln 2 . 2^n}{2n} \underset{\text{L'Hospital}}{=} \lim_{n \to \infty} \frac{(\ln 2)^2 . 2^n}{2} = \infty$

$n^2 \in O(2^n)$

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

c) $\lim_{n \to \infty} \frac{n \log n}{\sqrt{n}} = \lim_{n \to \infty} \sqrt{n} . \log n = \infty$

$\sqrt{n} \in O(n \log n)$ $n \log n > \sqrt{n}$

Scanned with CamScanner

d) $\lim_{n\to\infty} \frac{n.2^n}{3^n} = \lim_{n\to\infty} \frac{n}{\left(\frac{3}{2}\right)^n} \underset{\text{L'Hospital}}{\overset{=}{\downarrow}} \lim_{n\to\infty} \frac{1}{\ln\left(\frac{3}{2}\right).\left(\frac{3}{2}\right)^n}$

$= \lim_{n\to\infty} \frac{2^n}{\ln\left(\frac{3}{2}\right).3^n} \underset{\text{L'Hospital}}{\overset{=}{\downarrow}} \lim_{n\to\infty} \frac{\ln 2 . 2^n}{\ln\left(\frac{3}{2}\right)\ln 3 . 3^n} = \lim_{n\to\infty} \frac{\ln 2}{\ln\left(\frac{3}{2}\right).\ln 3} . \frac{2^n}{3^n}$

$= \frac{\ln 2}{\ln\left(\frac{3}{2}\right).\ln 3} . \frac{2^\infty}{3^\infty} = 0$ $\quad n.2^n \in O(3^n)$

$3^n > n.2^n$

e) $\lim_{n\to\infty} \frac{\sqrt{n+10}}{n^3} \underset{\text{L'Hospital}}{\overset{=}{\downarrow}} \lim_{n\to\infty} \frac{1}{6n^2\sqrt{n+10}} = 0$

$\sqrt{n+10} \in O(n^3)$ $\quad n^3 > \sqrt{n+10}$

4.

a) Basic operation is "if B[i,J] != B[J,i]" line.

b) $W(n) = \sum_{i=1}^{n-1} n-i$

As i increases in the outer loop. Number of iterations in inner loop decreases. Worst case of that algorithm is that there is no i and J such that B[i,J] != B[J,i]. Total $(n-1)+(n-2)+(n-3+\dots+2+1$ basic operations will be executed.

c) $W(n) = \sum_{i=1}^{n-1} n-i = (n-1)+(n-2)+(n-3)+\dots+1$

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

$$W(n) = \frac{n^2-n}{2} \in \Theta\left(\frac{n^2-n}{2}\right) = \Theta(n^2)$$

5.

a) Basic operation is "C[i,J]=C[i,J]+A[i,K]*B[K,J]" line

b) Since, there is no if statement or return inside the loops. Each loop will iterate n times. For that algorithm $T(n)=W(n)$.

$$T(n) = \sum_{i=1}^{n}\sum_{J=1}^{n}\sum_{k=1}^{n} 1$$

c) $$T(n) = \sum_{i=1}^{n}\sum_{J=1}^{n}\sum_{k=1}^{n} = \sum_{i=1}^{n}\sum_{J=1}^{n} n = \sum_{i=1}^{n} n^2 = n^3$$

$$T(n) = W(n) \in \Theta(n^3)$$

6. algorith3 (A[0..n-1], x) // x is desired number.

```
for i=0 to n-2 do
  for j=i+1 to n-1 do
    if(A[i]*A[j]==x)
      Print("(",A[i],",",A[j],")/n")
```

There is no beak condition for this algorithm.

$$T(n)=\sum_{i=1}^{n-1} n-i = (n-1)+(n-2)+(n-3)+\cdots+1$$

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

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

$$T(n) \in \Theta\left(\frac{n^2-n}{2}\right) = \Theta(n^2)$$

Emre Sezer

1901042640