DEVOIR 1

CSI2510

RAYOLD RAKOTONOMENJANAHARY

8884585

Question 1 :

a) $\log((n-2)(n+3))$ est $O(n^2)$?

$\log((n-2)(n+3)) = \log(n^2+n-6)$ et elle est définit $\forall n \geqslant 3$

$\forall n \geqslant 3$, $n^2+n-6 < n^2+n < n^2+n^2 = 2n^2$ d'où $\log(n^2+n-6) < \log(2n^2$

$\Rightarrow \log(2n^2) = \log(2) + 2\log(n)$, $\forall n \geqslant 3$ $\log(2) < n < n^2$, $\log(n) < n^2$

on a alors $\log(2) + 2\log(n) < n^2 + 2n^2 = 3n^2$

Avec $c = 3$ et $n_0 = 3$, nous vérifions que $\forall n \geqslant n_0$, $\log((n-2)(n+3)) \leqslant c$

Donc $\boxed{\log((n-2)(n+3)) \text{ est } O(n^2) \quad \text{Vrai}}$

b) $\log((n-2)(n+3))$ est $\Omega(n \log n)$?

On sait que $\log((n-2)(n+3))$ est définit $\forall n \geqslant 3$.

Par récurrence,

$$\log((n-2)(n+3)) = \log(n-2) + \log(n+3)$$

. Pour $n = 3$ $\quad \log(n-2) + \log(n+3) > n \log n$

$$\log(1) + \log(6) > 3 \log 3$$

Pour le premier terme on peut voir que $\log(n-2) + \log(n+3) > n \log n$

donc $\boxed{\log[(n-2)(n+3)] \text{ n'est pas } \Omega(n \log n). \text{ Faux}}$

c) $(n + \log(n))(n+8)$ est $O(n)$?

. $(n+\log(n))(n+8)$ est $O(n) \iff \exists c \in \mathbb{R}^{+*}, \exists n_0 \in \mathbb{N}^*$ tel que $\forall n \geqslant n_0$ $(n+\log(n))(n+8) \leq c \times n$

Développons d'abord $(n+\log(n))(n+8) = n^2 + 8n + n\log(n) + 8\log($

Par comparaison on peut tout de suite voir que :

$$n^2 + 8n + n\log(n) + 8\log(n) \geqslant c \cdot n, \ \forall n \geqslant 1$$

et $\forall c \in \mathbb{R}^{+*}$

Donc $\boxed{(n+\log(n))(n+8) \text{ est } O(n) \quad \text{FAUX}}$ car on ne trouve pas de c qui satisfait $f(n) \leq c \times n$

d) $(n+\log(n))(n+8)$ est $\Theta(n\log n)$?

$(n+\log(n))(n+8)$ est $\Theta(n\log n) \Leftrightarrow (n+\log(n))(n+8)$ est $O(n\log n)$ et $(n+\log(n))(n+8)$ est $\Omega(n\log n)$ mais on a vu dans la question précédente que $O(n\log n)$ n'est pas $O(n\log n)$ alors $(n+\log(n))(n+8)$ n'est pas $\Theta(n\log n)$

Réponse : FAUX.

e) $n^2+2n^2+3n^2+4n^2+\dots+n(n^2)$ est $O(n^2)$?

$n^2+2n^2+3n^2+4n^2+\dots+n(n^2)=(1+2+3+4+\dots+n)n^2$
$= \frac{n(n+1)}{2} n^2 = \frac{n^3(n+1)}{2}$ $\forall n \geq 1$, $\frac{n^3(n+1)}{2} > c.n^2$

donc $n^2+2n^2+3n^2+4n^2+\dots+n(n^2)$ n'est pas $O(n^2)$

Réponse : FAUX

f) $n^2+2n^2+3n^2+4n^2+\dots+n(n^2)$ est $\Omega(n^2)$?

On sait que $n^2+2n^2+3n^2+4n^2+\dots+n(n^2)=\frac{n^3(n+1)}{2}$ et elle est définit $\forall n \geq 1$. On peut voir alors que c'est évident que

$\frac{n^3(n+1)}{2} \geq c.n^2$ alors $n^2+2n^2+3n^2+4n^2+\dots+n(n^2)$

est $\Omega(n^2)$

Réponse : VRAI

g) $50n + 0{,}001n^3$ est $O(n^4)$?

$\forall n \geqslant 1$, $50n + 0{,}001n^3 \leqslant c \cdot n^4$

$50n \leqslant 50n^4$ et $0{,}001n^3 \leq 0{,}001n^4$ d'où $50n + 0{,}001n^3 \leq 50{,}001n^4$

alors $\forall n \geqslant 1$, et $c = 50{,}001$, $50n + 0{,}001n^3$ est $O(n^4)$

Réponse : Vrai

h) $50n + 0{,}001n^3$ est $\Omega(n^4)$?

$\forall n \geqslant 1$, $50n + 0{,}001n^3 \neq \Omega(n^4)$ car $50n + 0{,}001n^3 \not\geqslant c \cdot n^4$

et $\lim\limits_{n \to \infty} \dfrac{50n + 0{,}001n^3}{n^4} = 0$, il est impossible de trouver

un $c > 0$ pour satisfaire $\Omega(n^4)$.

Réponse : FAUX

Question 2 :

a) Le nombre de décalage de bits effectués est le nombre de passage de l'algorithme à la ligne 6 multiplier par 2 (on multiplie par 2 car il y a le nombre de décalage à droite et le nombre de décalage à gauche). Il faut noter que ~~...~~ le nombre de décalage dépend de la valeur de y. décimal

Le nombre y est représenté sur n bit en binaire, on a la formule : $2^n - 1 = y$. n dans notre cas représente le nombre de décalage de y, alors

$$2^n = y+1$$

$$\log(2^n) = \log(y+1) \Rightarrow n\log(2) = \log(y+1)$$

donc le nombre de décalage à gauche à la ligne 8 est $n = \dfrac{\log(y+1)}{\log(2)}$

de même à la ligne 9, nbre décalage à droite $= \dfrac{\log(y+1)}{\log(2)}$

Donc le nombre de décalage total = nbre décalage à droite + nbre de décalage à gauche

$$= \frac{\log(y+1)}{\log(2)} + \frac{\log(y+1)}{\log(2)}$$

$$= \boxed{2\,\frac{\log(y+1)}{\log(2)}}$$

b) L'ordre de grandeur de la complexité dépend du nombre de décalage total donc on a

$$\boxed{O(\log(y))}$$

Question 3 :

a) n :

Ancienne machine :

$1s \longrightarrow n$

Nouvelle machine

$\frac{1}{10} s \longrightarrow n$

donc on a $\boxed{10\,n}$ de données pour 1 seconde dans la nouvelle machine

b) n^2 :

Nouvelle machine

- $\frac{1}{10} s \longrightarrow n$

$\Rightarrow \frac{1}{10} \cdot \frac{1}{10} s \longrightarrow n^2$

$1 s \longrightarrow ?$

donc on a $\boxed{100\,n^2}$ de données pour 1 seconde.

c) $\log n$

- $\frac{1}{10} s \longrightarrow n$

$\log\left(\frac{1}{10}\right) \longrightarrow \log n$

$1s \longrightarrow ?$

donc on a $\boxed{-\frac{\log n}{\log 10}}$ données pour 1 secondes

Question 4:

a) i) Le tableau contient des nombres binaires (0 ou 1) avec $n=2$, l'ordre de complexité est $O(N)$

ii) n est à peu près égal à N alors, on a l'ordre de complexité $O(N^2)$

b) Un algorithme plus efficace:

```
for ( i = 0 ; i < n ; i++ ) {
    H[i] = 0 ;
    for (j = 0 ; j ≤ N ; j++ ) {
        if ( A[j] = = i )
            H[i]++ ;
}
```

On peut combiner les deux boucles for

Donc on a :

```
for (i = 0 ; i < N ; i++ ) {
        H[i] = 0 ;
        H[A[i]] ++ ;
  }
```

La nouvelle complexité de l'algorithme pour les deux cas des questions précédentes : i) $O(n)$
ii) $O(n)$