SOLUTIONS[1] SÉRIE 2 (chapitre 2a)

Question # 1

Utilisez la notation la plus appropriée parmi O, Ω et Θ pour indiquer la classe d'efficacité (par rapport au temps) de la recherche séquentielle ci-dessous dans les différents cas.

Algorithme 1 : RechercheSequentielle($A[0..n-1], d$)

```
pour i = 0..n − 1 faire
    si A[i] = d alors
        retourner i
retourner −1
```

A) *Pire cas*

Solution :

Puisque $C_{worst}(n) = n$, on a que $C_{worst}(n) \in \Theta(n)$.

B) *Meilleur cas*

Solution :

Puisque $C_{best}(n) = 1$, on a que $C_{best}(n) \in \Theta(1)$.

C) *Cas moyen*

Solution :

Aucune distribution de probabilité n'est donné sur les instances possibles. Nous allons donc en définir une. Soit p la probabilité que l'élément recherché d est dans le vecteur A. Dans le cas où l'élément recherché est dans le vecteur A, la probabilité que l'élément se retrouve à la position i est uniforme, soit $\frac{1}{n}$.

$$C_{avg}(n) = \sum_{x:|x|=n} P(x)C(x) \tag{1}$$

Il y a $n+1$ instances de taille n. Une instance pour chaque position dans le vecteur où l'élément recherché peut se trouver, plus une instance où l'élément recherché n'est pas dans le vecteur.

$$= \sum_{i=0}^{n-1} P(A[i] = d)C(A[i] = d) + P(d \notin A)C(d \notin A) \tag{2}$$

$$= \sum_{i=0}^{n-1} \frac{p}{n}(i+1) + (1-p)n \tag{3}$$

$$= \frac{p}{n} \sum_{i=1}^{n} i + (1-p)n \tag{4}$$

$$= \frac{p}{n} \frac{n(n+1)}{2} + (1-p)n \tag{5}$$

$$= p\frac{n+1}{2} + (1-p)n \tag{6}$$

1. Dans le but d'alléger les démonstrations, nous utilisons $f(n) \leq cg(n)$ comme définition de $f(n) \in O(g(n))$. Nous traitons implicitement le fait qu'il existe une certaine constante c et que l'inégalité est vraie pour tout $n \geq n_0$ pour un certain n_0. Nous nous permettons de faire cette simplification car l'idée des preuves se situe rarement à l'intérieur des constantes c et n_0.

Question # 2

En vous appuyant uniquement sur notre analyse de l'algorithme d'Euclide effectuée au chap 1, exprimez l'efficacité de l'algorithme d'Euclide en pire cas à l'aide de la notation asymptotique.

Solution :

Rappel : la taille s de l'instance est le nombre de bits utilisé par les deux entiers (m,n) à l'entrée de l'algorithme d'Euclide. Notre analyse du chap 1 nous a donné $C_{worst}(s) \in O(s)$. Alors $C(s) \in O(s)$. Le(s) pire cas n'a pas été identifié et nous n'avons pu conclure que $C_{worst}(s) \in \Theta(s)$. Ainsi notre analyse ne nous a pas permis de conclure que la borne asymptotique supérieure est optimale.

Question # 3
Démontrez la règle du maximum pour Ω.
Solution :

$$
\begin{aligned}
& t_1(n) \in \Omega(g_1(n)) \wedge t_2(n) \in \Omega(g_2(n)) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } \Omega \ \rangle \\
& t_1(n) \geq c_1 g_1(n) \wedge t_2(n) \geq c_2 g(n) \\
\Rightarrow & \ t_1(n) + t_2(n) \geq c_1 g_1(n) + c_2 g_2(n) \\
\Rightarrow & \qquad \langle \text{ Posons } c_3 = min(c_1, c_2) \ \rangle \\
& t_1(n) + t_2(n) \geq c_3(g_1(n) + g_2(n)) \\
\Rightarrow & \qquad \langle \text{ pour tout } a, b \geq 0 \text{, on a } a + b \geq max(a, b). \ \rangle \\
& t_1(n) + t_2(n) \geq c_3 max(g_1(n), g_2(n)) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } \Omega \ \rangle \\
& t_1(n) + t_2(n) \in \Omega(max(g_1(n), g_2(n)))
\end{aligned}
$$

Question # 4
Démontrez la règle du maximum pour Θ.
Solution :
Résultat 1 :

$$
\begin{aligned}
& t_1(n) \in \Theta(g_1(n)) \wedge t_2(n) \in \Theta(g_2(n)) \\
\Rightarrow & \qquad \langle \text{ car } \Theta(g(n)) = \Omega(g(n)) \cap O(g(n)) \ \rangle \\
& t_1(n) \in \Omega(g_1(n)) \wedge t_2(n) \in \Omega(g_2(n)) \\
\Rightarrow & \qquad \langle \text{ en vertu de la règle du maximum pour } \Omega \ \rangle \\
& t_1(n) + t_2(n) \in \Omega(max(g_1(n), g_2(n)))
\end{aligned}
$$

Résultat 2 :

$$
\begin{aligned}
& t_1(n) \in \Theta(g_1(n)) \wedge t_2(n) \in \Theta(g_2(n)) \\
\Rightarrow & \qquad \langle \text{ car } \Theta(g(n)) = \Omega(g(n)) \cap O(g(n)) \ \rangle \\
& t_1(n) \in O(g_1(n)) \wedge t_2(n) \in O(g_2(n)) \\
\Rightarrow & \qquad \langle \text{ en vertu de la règle du maximum pour } O \ \rangle \\
& t_1(n) + t_2(n) \in O(max(g_1(n), g_2(n)))
\end{aligned}
$$

Ces deux résultats impliquent que :

$$t_1(n) + t_2(n) \in \Omega(max(g_1(n), g_2(n))) \cap O(max(g_1(n), g_2(n))) = \Theta(max(g_1(n), g_2(n)))$$

Question # 5

Démontrez que $f(n) \in \Theta(g(n)) \Leftrightarrow \Theta(f(n)) = \Theta(g(n)) \Leftrightarrow O(f(n)) = O(g(n))$

Solution :

- $f(n) \in \Theta(g(n)) \Rightarrow \Theta(f(n)) = \Theta(g(n))$ *:*

$$
\begin{array}{ll}
 & \text{Soit } x(n) \in \Theta(f(n)) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } \Theta \ \rangle \\
 & c_1 f(n) \leq x(n) \leq c_2 f(n) \\
\Rightarrow & \qquad \langle\ f(n) \in \Theta(g(n))\ \rangle \\
 & d_1 g(n) \leq c_1 f(n) \leq x(n) \leq c_2 f(n) \leq d_2 g(n) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } \Theta \ \rangle \\
 & x(n) \in \Theta(g(n))
\end{array}
$$

De la même façon (et en utilisant la symétrie de Θ*), on peut montrer que* $x(n) \in \Theta(g(n)) \Rightarrow x(n) \in \Theta(f(n))$ *sous l'hypothèse que* $f(n) \in \Theta(g(n))$*. On obtient alors l'égalité entre* $\Theta(f(n))$ *et* $\Theta(g(n))$

- $\Theta(f(n)) = \Theta(g(n)) \Rightarrow O(f(n)) = O(g(n))$ *:*

$$
\begin{array}{ll}
 & \text{Soit } x(n) \in O(f(n)) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } O \ \rangle \\
 & x(n) \leq c f(n) \\
\Rightarrow & \qquad \langle\ f(n) \in \Theta(g(n)) \text{ car } \Theta(f(n)) = \Theta(g(n))\ \rangle \\
 & x(n) \leq c f(n) \text{ et } d_1 g(n) \leq f(n) \leq d_2 g(n) \\
\Rightarrow & \\
 & x(n) \leq c \times d_2 g(n) \\
\Leftrightarrow & \qquad \langle \text{ Définition de } O \ \rangle \\
 & x(n) \in O(g(n))
\end{array}
$$

De la même façon, on peut montrer que $x(n) \in O(g(n)) \Rightarrow x(n) \in O(f(n))$ *sous l'hypothèse que* $\Theta(f(n)) = \Theta(g(n))$*. On obtient alors l'égalité entre* $O(f(n))$ *et* $O(g(n))$

- $O(f(n)) = O(g(n)) \Rightarrow f(n) \in \Theta(g(n))$ *:*

  En assumant $O(f(n)) = O(g(n))$ *et par la réflexivité de* O*, on a que* $f(n) \in O(g(n))$ *et* $g(n) \in O(f(n))$*. On déduit donc que* $f(n) \leq d_1 g(n)$ *et* $\frac{1}{d_2} g(n) \leq f(n)$ *ce qui donne bien* $f(n) \in \Theta(g(n))$*.*

Question # 6

Démontrez que :

$$\begin{aligned} f(n) \in O(g(n)) \wedge f(n) \notin \Theta(g(n)) &\Leftrightarrow f(n) \in O(g(n)) \wedge f(n) \notin \Omega(g(n)) \\ &\Leftrightarrow f(n) \in O(g(n)) \wedge g(n) \notin O(f(n)) \\ &\Leftrightarrow O(f(n)) \subset O(g(n)) \end{aligned}$$

Solution :

- *$f(n) \in O(g(n)) \wedge f(n) \notin \Theta(g(n)) \Rightarrow f(n) \in O(g(n)) \wedge f(n) \notin \Omega(g(n))$:*
  Assumons $f(n) \in O(g(n)) \wedge f(n) \notin \Theta(g(n))$ et supposons que $f(n) \in \Omega(g(n))$. On a donc que $f(n) \leq cg(n)$ et $f(n) \geq dg(n)$, ce qui donne que $dg(n) \leq f(n) \leq cg(n)$ et donc $f(n) \in \Theta(g(n))$. Ceci est une contradiction, on a donc $f(n) \notin \Omega(g(n))$.

- *$f(n) \in O(g(n)) \wedge f(n) \notin \Omega(g(n)) \Rightarrow f(n) \in O(g(n)) \wedge g(n) \notin O(f(n))$:*
  Assumons $f(n) \in O(g(n)) \wedge f(n) \notin \Omega(g(n))$ et supposons $g(n) \in O(f(n))$. Par la dualité entre O et Ω, on obtient que $f(n) \in \Omega(g(n))$. Ceci est une contradiction, on a donc $g(n) \notin O(f(n))$.

- *$f(n) \in O(g(n)) \wedge g(n) \notin O(f(n)) \Rightarrow O(f(n)) \subset O(g(n))$:*
  Comme $f(n) \in O(g(n))$, on a que $O(f(n)) \subseteq O(g(n))$ (pour tout $x(n) \in O(f(n))$, on a $x(n) \leq c(f(n)) \leq d(g(n))$ et donc $x(n) \in O(g(n))$). Si $O(f(n)) = O(g(n))$, alors forcément $g(n) \in O(f(n))$(car $g(n) \in O(g(n))$). Ceci est une contradiction. Puisque $O(f(n)) \subseteq O(g(n))$, mais que $O(f(n)) \neq O(g(n))$, on obtient que $O(f(n)) \subset O(g(n))$.

- *$O(f(n)) \subset O(g(n)) \Rightarrow f(n) \in O(g(n)) \wedge f(n) \notin \Theta(g(n))$:*
  Comme $O(f(n)) \subset O(g(n))$, il est clair que $f(n) \in O(g(n))$. Par la question 5, on sait que si $f(n) \in \Theta(g(n))$, on obtient que $O(f(n)) = O(g(n))$, ce qui est une contradiction. On doit donc avoir $f(n) \notin \Theta(g(n))$.

Question # 7

Soit deux fonctions positives asymtotiquement $f(n)$ et $g(n)$. Les relations $=$, $\neq$, $\subset$ et $\not\subset$ sont quatre relations intéressantes pour comparer les ensembles $O(f(n))$ et $O(g(n))$.

A) Pourquoi n'est-il pas intéressant d'utiliser seulement $=$ ou bien $\subset$ pour comparer les ensembles $\Theta(f(n))$ et $\Theta(g(n))$?

Solution :

Il n'est pas possible d'avoir $\Theta(f(n)) \subset \Theta(g(n))$. On peut facilement montrer que si il existe une fonction x telle que $x \in \Theta(f(n))$ et $x \in \Theta(g(n))$, alors on a $\Theta(f(n)) = \Theta(g(n))$.

B) Que devrait-on utiliser pour comparer $\Theta(f(n))$ et $\Theta(g(n))$?

Solution :

On devrait utiliser $=$ ou bien $\neq$.

***Question # 8**

Pour chacune des fonctions suivantes, indiquez la classe $\Theta(g(n))$ *à laquelle la fonction appartient (Utilisez la forme la plus simple possible pour* $g(n)$*). Démontrez vos affirmations.*

A) $(n^2+1)^{10}$

Solution :

$(n^2+1)^{10} \in \Theta(n^{20})$

$$\lim_{n\to\infty} \frac{(n^2+1)^{10}}{n^{20}} = \lim_{n\to\infty} \frac{(n^2+1)^{10}}{(n^2)^{10}} = \lim_{n\to\infty} (\frac{(n^2+1)}{n^2})^{10} = \lim_{n\to\infty} (1+\frac{1}{n^2})^{10} = 1$$

B) $\sqrt{10n^2+7n+3}$

Solution :

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

$$\lim_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{n} = \lim_{n\to\infty} \sqrt{\frac{10n^2+7n+3}{n^2}} = \lim_{n\to\infty} \sqrt{10+\frac{7}{n}+\frac{3}{n^2}} = \sqrt{10}$$

C) $2n\lg(n+2)^2 + (n+2)^2\lg(\frac{n}{2})$

Solution :

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

$2n\lg(n+2)^2 + (n+2)^2\lg(\frac{n}{2})$

$=$

$2n2\lg(n+2) + (n+2)^2(\lg n - 1)$

$\in$ $\langle$ *par la règle du maximum* $\rangle$

$\Theta((n+2)^2(\lg n-1))$

$=$

$\Theta((n^2+4n+4)(\lg n-1))$

$=$

$\Theta(n^2\lg n + 4n\lg n + 4\lg n - n^2 - 4n - 4)$

$=$

$\langle \lim_{n\to\infty} \frac{n^2\lg n+4n\lg n+4\lg n-n^2-4n-4}{n^2\lg n} = 1$, *voir plus bas.* $\rangle$

$\Theta(n^2\lg n)$

$\lim_{n\to\infty} \frac{n^2\lg n+4n\lg n+4\lg n-n^2-4n-4}{n^2\lg n}$

$=$

$\lim_{n\to\infty} \frac{n^2\lg n}{n^2\lg n} + \frac{4n\lg n}{n^2\lg n} + \frac{4\lg n}{n^2\lg n} - \frac{n^2}{n^2\lg n} - \frac{4n}{n^2\lg n} - \frac{4}{n^2\lg n}$

$=$

$\lim_{n\to\infty} 1 + \frac{4}{n} + \frac{4}{n^2} - \frac{1}{\lg n} - \frac{4}{n\lg n} - \frac{4}{n^2\lg n}$

$=$

$1 + \frac{4}{\infty} + \frac{4}{\infty} - \frac{1}{\infty} - \frac{4}{\infty} - \frac{4}{\infty}$

$=$

$1+0+0-0-0-0=1$

D) $2^{n+1} + 3^{n-1}$

Solution :

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

$$\begin{aligned} & 2^{n+1} + 3^{n-1} \\ = \quad & \\ & 2^n 2 + 3^n \tfrac{1}{3} \\ \in \quad & \qquad \langle \textit{ par la règle du maximum } \rangle \\ & \Theta(3^n) \end{aligned}$$

E) $\lfloor \lg n \rfloor$

Solution :

Puisque $x - 1 < \lfloor x \rfloor \leq x$*, on obtient que :*

$$\begin{aligned}
\lfloor \lg n \rfloor &\leq \lg n = \lg(a)\log_a(n) \Rightarrow \lfloor \lg n \rfloor \in O(\log n) \\
\lfloor \lg n \rfloor &> \lg n - 1 \\
&= \lg(a)\log_a(n) - 1 \\
&\geq \lg(a)\log_a(n) - \frac{1}{2}\lg(a)\log_a(n) && \forall n \geq 2 \\
&= \frac{1}{2}\lg(a)\log_a(n) \\
&\Longrightarrow \lfloor \lg n \rfloor \in \Omega(\log n)
\end{aligned}$$

Alors $\lfloor \lg n \rfloor \in \Theta(\log n)$.

***Question # 9**

Placez les fonctions suivantes en ordre croissant de leur ordre de croissance.

$$(n-2)!, 5\lg(n+100)^{10}, 2^{2n}, 0.001n^4 + 3n3 + 1, (\ln n)^2, \sqrt[3]{n}, 3^n$$

Solution :

$$5\lg(n+100)^{10}, (\ln n)^2, \sqrt[3]{n}, 0.001n^4 + 3n3 + 1, 3^n, 2^{2n}, (n-2)!$$

Question # 10

Les valeurs contenus dans le tableau 2.1 (voir A. Levitin) suggère que les fonctions suivantes sont placées en ordre croissant de leur ordre de croissance :

$$\lg n, n, n\lg n, n^2, n^3, 2^n, n!$$

A) *Est-ce que les valeurs du tableau forme une preuve que les fonctions sont placées en ordre croissant de leur ordre de croissance ?*

Solution :

L'ordre de croissance d'une fonction est étroitement lié au comportement asymptotique de cette fonction (lorsque n tends vers l'infini). Donc, en aucun cas, un ensemble fini de valeurs d'une fonction peut constituer une preuve de l'ordre de croissance de cette fonction.

B) *Démontrez que les fonctions ci-haut sont listées en ordre croissant de leur ordre de croissance.*
Solution :

$$\begin{aligned}
\lim_{n\to\infty} \frac{\lg n}{n} &= \lim_{n\to} \frac{1}{n} \lg e = \lim_{n\to\infty} \frac{c}{n} = 0 \\
\lim_{n\to\infty} \frac{n}{n \lg n} &= \lim_{n\to\infty} \frac{1}{\lg n} = 0 \\
\lim_{n\to\infty} \frac{n \lg n}{n^2} &= \lim_{n\to\infty} \frac{\lg n}{n} \\
&= 0 && \textit{Déjà démontré ci-dessus} \\
\lim_{n\to\infty} \frac{n^2}{n^3} &= \lim_{n\to\infty} \frac{1}{n} = 0 \\
\lim_{n\to\infty} \frac{n^3}{2^n} &= \lim_{n\to\infty} \frac{3n^2}{2^n \ln 2} && \textit{Règle de L'Hôpital} \\
&= \lim_{n\to\infty} \frac{6n}{2^n (\ln 2)^2} && \textit{Règle de L'Hôpital} \\
&= \lim_{n\to\infty} \frac{6}{2^n (\ln 2)^3} && \textit{Règle de L'Hôpital} \\
&= 0 \\
\lim_{n\to\infty} \frac{2^n}{n!} &= 0 && \textit{Voir la solution de \#3B série 0}
\end{aligned}$$

Question # 11
Démontrez que tout polynôme $p(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0$ *tel que* $a_k > 0$ *appartient à* $\Theta(n^k)$.
Solution :

$$\lim_{n\to\infty} \frac{a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0}{n^k} = \lim_{n\to\infty} a_k + \frac{a_{k-1}}{n} + \ldots + \frac{a_0}{n^k} = a_k > 0$$

Question # 12
Démontrez que deux fonctions exponentielle a^n *et* b^n *ayant des bases différentes* ($a \neq b$) *ont des ordre de croissance différente.*
Solution :

$$\lim_{n\to\infty} \frac{a^n}{b^n} = \lim_{n\to\infty} (\frac{a}{b})^n = \begin{cases} 0 & \text{si } a < b \\ \infty & \text{si } a > b \end{cases}$$

Question # 13
Démontrez que, pour tout $a \geq 2$ *et* $b \geq 2$, *on a :* $\log_b(n) \in \Theta(\log_a(n))$ *(et alors* $\log_a(n) \in \Theta(\log_b(n))$*).*
Solution :

$$\lim_{n\to\infty} \frac{\log_a(n)}{\log_b(n)} = \lim_{n\to\infty} \frac{\log_a(b) \log_b(n)}{\log_b(n)} = \log_a(b)$$

Question # 14

1. ***E**lvis, **Ji**mmy et **Ja**nis analysent un même algorithme* $\mathbf{A}$ *et obtiennent les résultats* $T_E(n), T_i(n)$ *et* $T_a(n)$ *respectivement. Elvis obtient* $T_E(n) \in O(n^2)$ *en pire cas. Jimmy obtient* $T_i(n) \in \Omega(n^2)$ *en meilleur cas. Janis affirme que* $T_a(n) \in \theta(n^2)$ *pour tous les cas. Si Elvis et Jimmy ont raison, le résultat de Janis est-il correct ? Justifiez.*

2. *Soient* $\mathbf{A}$ *et* $\mathbf{B}$ *des algorithmes prenant des temps, en meilleur cas, dans* $\Omega(n^2)$ *et* $\Omega(n \log n)$ *respectivement. Est-il possible qu'une implantation de l'algorithme* $\mathbf{A}$ *soit plus efficace qu'une implantation de l'algorithme* $\mathbf{B}$ *sur tous les exemplaires ? Justifiez.*
3. *Alice et Bob analysent un même algorithme* $\mathbf{A}$ *et obtiennent les résultats* $T_A(n)$ *et* $T_B(n)$ *respectivement. Alice obtient que* $T_A(n) \in \theta(n^3)$ *en pire cas. Bob obtient que* $T_B(n) \in \Omega(n^2 \log n^2)$ *pour tous les cas. Le résultat d'Alice implique-t-il celui de Bob ? Justifiez.*
4. *Énoncez* le principe d'invariance *en algorithmique.*

Solutions :

1. ***Oui****. Elvis* $\Rightarrow T(n) \in O(n^2)$ *dans tous les cas et Jimmy* $\Rightarrow T(n) \in \Omega(n^2)$ *dans tous les cas, d'où le résultat.*
2. ***Oui****. Exemple :* $T_A(n) = n^2$ *et* $T_B(n) = 5n^3 + 5$
3. **Non**. *On pourrait avoir, par exemple,* $T(n) = n^2$ *(en meilleur cas).*
4. *Les* **temps d'exécution** *de* **deux implantations** *d'un* **même algorithme** *ne diffèrent qu'à une* **constante multiplicative** *près (voir notes de cours chapitre 2).*

***Question # 15**

Soient A, B, C, D *et* E *cinq algorithmes qui résolvent le même problème* P *et pour lesquels nous avons obtenu les résultats d'analyse suivants respectivement :*

1. $T_A(n) \in O(n^2)$ *en pire cas ;*
2. $T_B(n) \in \Omega(n^2)$ *dans tous les cas ;*
3. $T_C(n) \in \Theta(n)$ *en meilleur cas ;*
4. $T_D(n) \in \Theta(n^2 \lg n)$ *dans tous les cas ;*
5. $T_E(n) \in \Theta(n^2)$ *en pire cas.*

a) *Vous êtes à concevoir un algorithme* Z *qui résout un problème* P' *ayant des instances de taille* n. *Votre algorithme* Z *ne doit jamais prendre plus de* $n^4 \lg n$ *unités de temps (à une constante près et pour des entrées de taille assez grande). Dans* Z, *il y a* $n^2 \lg n$ *appels à* Résoudre le problème P *(tous sur des instances de taille* n*). Donnez, parmi les cinq algorithmes ci-haut, tous ceux qui peuvent être utilisés dans votre algorithme* Z, *sans dépasser la limite de temps.*

b) *Vous êtes à élaborer un système qui doit fonctionner en temps réel (le facteur temps est primordial). Pour ce faire, vous devez choisir un (et un seul) algorithme parmi les 5 ci-haut. Quel est, selon vous, le choix le plus prometteur ? Justifiez.*

c) *Pour chacun des 5 algorithmes ci-haut, dites, avec justifications, si on est assuré que son temps d'exécution* ***en pire cas*** *appartient à l'ensemble suivant :*

$$S = (O(n^3) \cap \Omega(n^2)) \cup \Theta(n).$$

d) *En supposant que les analyses 2 et 3 soient exactes, est-il possible que les algorithmes* B *et* C *soient en fait un seul et même algorithme ? Justifiez.*

Solutions :

a) *Les algorithmes* A *et* E *peuvent être utilisés.*

b) *L'analyse des algorithmes B et C ne donne aucune informations sur le maximum de temps requis pour ces algorithmes. Ils peuvent donc possiblement être extrêmement lent. Clairement, le choix de l'algorithme D est moins bon que A et E. Puisque E nous assure un temps de n^2 en pire cas, alors que A nous assure un temps inférieur ou égal à n^2 en pire cas, l'algorithme E ne peut être meilleur que A. A est donc le choix le plus prometteur.*

c) *A : non, le temps en PC est au plus n^2 mais pourrait être inférieur, $n \lg n$ par exemple et dans ce cas il n'est pas dans l'ensemble car $n \lg n \notin \Omega(n^2)$ et $n \lg n \notin \Theta(n)$.*
   B : non, aucune borne supérieur sur le temps en PC, ce dernier peut donc être n^{10} qui n'appartient clairement pas à S.
   C : non, même explication que B.
   D : oui, comme $T_D(n) \in \Theta(n^2 \lg n)$, on sait que $T_D(n) \in \Omega(n^2 \lg n) \subset \Omega(n^2)$ et $T_D(n) \in O(n^2 \lg n) \subset O(n^3)$.
   E : oui, comme $T_E(n) \in \Theta(n^2)$, on sait que $T_E(n) \in \Omega(n^2)$ et $T_E(n) \in O(n^2) \subset O(n^3)$.

d) *Non car il y a une contradiction. L'analyse 2 nous dit que $T(n)$ est borné inférieurement par n^2 dans tous les cas (en particulier pour le meilleur cas). Par contre, l'analyse 3 nous dit que $T(n)$ est exactement n en meilleur cas. Mais comme n ne peut être plus petit que n^2, il ne peut donc pas s'agir du même algorithme.*

***Question # 16**

Déterminez si les énoncés suivants sont vrais ou faux. Justifiez en mots (ou en montrant les étapes de votre démarche si nécessaire).

A) $376n + 98 \in O(n)$
   Solution :
   Vrai. $O(n)$ est l'ensemble des fonctions dont l'ordre de croissance est plus petit ou égal à n.
   Par la régle du maximum on laisse tomber la constante 98.
   De plus, la constante 376 n'affecte pas l'ordre de grandeur de la fonction.
   Donc, $376n + 98 \in O(n)$.

B) $2n^2 + n \in \Omega(n)$
   Solution :
   Vrai. $\Omega(n)$ est l'ensemble des fonctions dont l'ordre de croissance est plus grand ou égal à n.
   En utilisant la limite du ratio $\frac{2n^2+n}{n}$ lorsque n tend vers l'infini pour comparer $2n^2 + n$ et n, nous obtenons ∞.
   De ce fait, $2n^2 + n \in \Omega(n)$.

C) $\ln(n^n) + \log_2 n \in \Theta(\ln n)$
   Solution :
   Faux. $\Theta(\ln n)$ est l'ensemble des fonctions dont l'ordre de croissance est $\ln n$.
   Par la régle du maximum on laisse tomber la fonction $\log_2 n$.
   Or, par l'arithmétique des logs nous avons que $\ln(n^n) = n \ln(n)$.
   Donc, $\ln(n^n) + n \notin \Theta(\ln n)$.