# SÉRIE 5 (Chapitre 5)

**Question # 1**

*Soit $A[0 \ldots n-1]$, un tableau de $n$ éléments comparables (Pour simplifier, on suppose ici que les élément sont tous distincts). Dans le tri par insertion (voir Algorithme 1), une paire d'indices $(i, j)$ est une inversion si $i < j$ et $A[i] > A[j]$.*

**Algorithme 1 :** TriParInsertion($A[0..n-1]$)

```
1 pour i = 1..n − 1 faire
2     v ← A[i] // Élément à insérer
3     j ← i − 1 // Positions possibles d'insertion
4     tant que j ≥ 0 ∧ A[j] > v faire
5         A[j + 1] ← A[j]
6         j ← j − 1
7     A[j + 1] ← v
8 retourner A
```

A) *Décrivez le tableau ayant le plus grand nombre d'inversions pour une taille $n$ donnée. Quel est ce nombre d'inversions ?*

B) *Décrivez le tableau ayant le plus petit nombre d'inversions pour une taille $n$ donnée. Quel est ce nombre d'inversions ?*

C) *Analysez l'efficacité de l'algorithme en cas moyen en prenant la comparaison «$A[j] > v$» comme opération de base. Supposez que tous les éléments de $A$ sont distincts.*

**Question # 2**

*Écrivez un algorithme non-récursif qui résout le problème de sélection en se basant sur les partitions. Le problème de sélection est celui de trouver le $k$ième plus petit élément dans un tableau de $n$ nombres.*

**Question # 3**

*Vous avez un ensemble de pièces d'or $P[1, \ldots, n]$. Toutes ces pièces sont identiques à l'exception d'une unique fausse pièce, qui est supposée plus légère que les autres. Vous avez à votre disposition une balance pouvant uniquement comparer deux poids l'un contre l'autre. Vous pouvez mettre plusieurs pièces sur la balance. Par conséquent, l'algorithme Balance($A$,$B$) retourne 0 si les poids sont égaux et $-1$ si $A$ est plus léger que $B$ et 1 si $A$ est plus lourd que $B$. Écrivez un algorithme qui trouve la fausse pièce avec le minimum d'appel à l'algorithme Balance. Votre algorithme doit être en $\Theta(\log(n))$. Faites l'analyse de l'efficacité de votre algorithme.*

**Question # 4**

*Un groupe de personnes se réunissent à l'occasion d'une oeuvre de charité. Vous êtes en charge de remettre un colis à une personne selon les indications suivantes :*

1. *elle ne connaît personne parmi les invités ;*
2. *elle est connue de tous les invités.*

*Votre mission est de trouver le destinataire du colis en posant le moins de questions possibles aux invités, histoire de ne pas trop vous faire remarquer. Écrivez un algorithme trouvant la personne cherchée. Vous pouvez faire appel à l'algorithme Connait($A$,$B$) qui retourne* vrai *si $A$ connaît $B$ et* faux *sinon.*

**Question # 5**

*Vous devez analyser l'algorithme* Mystère1*. Si la complexité de l'algorithme varie parmi les instances de même taille, identifiez le pire et le meilleur cas. Procédez ensuite à une analyse en pire cas, en meilleur cas et en cas moyen. Si la complexité de l'algorithme est la même pour toutes les instances d'une même taille, alors donnez la complexité de cet algorithme en fonction de $n$. Toutes vos réponses doivent utiliser la notation $\Theta$.*

**Algorithme 2 :** Mystère1($A[0..n-1]$)

```
1 ρ ← Uniforme(1, n)
2 c ← 0
3 pour i from 1 to nρ faire
4 |  c ← c + A[i mod n]
5 retourner c
```

**Question # 6**

*Vous devez analyser l'algorithme* Mystère2*. Si la complexité de l'algorithme varie parmi les instances de même taille, identifiez le pire et le meilleur cas. Procédez ensuite à une analyse en pire cas, en meilleur cas et en cas moyen. Si la complexité de l'algorithme est la même pour toutes les instances d'une même taille, alors donnez la complexité de cet algorithme en fonction de $n$. Toutes vos réponses doivent utiliser la notation $\Theta$. Vous pouvez supposer que la probabilité d'avoir $i$ bits d'allumés (pour $0 \leq i \leq n$) dans la séquence $B$ est de $\frac{1}{n+1}$.*

**Algorithme 3 :** Mystère2($B[0..n-1]$)

```
1 // B est un tableau de bits : B[i] ∈ {0,1}, ∀i ∈ {0,...,n-1}
2 s ← 0
3 pour i from 0 to n − 1 faire
4 |  s ← s + B[i]
5 ρ ← Uniforme(1, n)
6 c ← 0
7 pour i from 1 to sρ faire
8 |  c ← c + B[i mod n]
9 retourner c
```

**Question # 7**

*Vous devez analyser l'algorithme* Mystère3*. Si la complexité de l'algorithme varie parmi les instances de même taille, identifiez le pire et le meilleur cas. Procédez ensuite à une analyse en pire cas, en meilleur cas et en cas moyen. Si la complexité de l'algorithme est la même pour toutes les instances d'une même taille, alors donnez la complexité de cet algorithme en fonction de $n$. Toutes vos réponses doivent utiliser la notation $\Theta$. Vous pouvez supposer que la probabilité d'avoir $i$ bits d'allumés (pour $0 \leq i \leq n$) dans la séquence $B$ est de $\frac{1}{n+1}$.*

**Algorithme 4 :** Mystère3($B[0..n-1]$)

```
1  // B est un tableau de bits : B[i] ∈ {0,1}, ∀i ∈ {0,...,n−1} ;
2  s ← 0
3  pour i from 0 to n − 1 faire
4      s ← s + B[i]
5  a ← Uniforme(3, n)
6  c ← 0
7  si s = 0 alors
8      pour i from 0 to n − 1 faire
9          c ← c + B[i mod n]
10 sinon si 0 < s ≤ ⌊n/2⌋ alors
11     pour i from 1 to sa faire
12         c ← c + B[i mod n]
13 sinon
14     c ← 4
15 retourner c
```