

## Examen Algorithmique Avancée

Les calculatrices et autres appareils électroniques sont interdits. Les téléphones mobiles doivent être éteints et rangés dans les sacs.

### Exercice 1.

---

**Question 1)** Montrer en détaillant toutes les étapes que  $7n + \sqrt{n} + 15 \in \Theta(n)$ , avec  $n \in \mathbb{R}^{**}$ .

**Question 2)** Etant donné deux fonctions  $f$  et  $g$  sur  $\mathbb{R}^{**}$ , montrer que  $\max\{f, g\} \in \Theta(f + g)$ .

**Question 3)** Vous devez choisir parmi ces trois algorithmes suivants; Quelle est la complexité de chacun ? Lequel est le plus rapide?

- A. L'algorithme **A** résout les problèmes de taille  $n$  en les divisant en cinq sous-problèmes de taille  $n/2$ , il résout les sous-problèmes récursivement, et il combine les solutions en temps  $O(n)$ .
- B. L'algorithme **B** résout les problèmes de taille  $n$  en résolvant récursivement deux sous-problèmes de taille  $n-1$ , il résout les sous-problèmes récursivement, et il combine les solutions en temps  $O(1)$ .
- C. L'algorithme **C** résout les problèmes de taille  $n$  en les divisant en neuf sous-problèmes de taille  $n/3$ , il résout les sous-problèmes récursivement, et il combine les solutions en temps  $O(n^2)$ .

### Exercice 2.

---

Soit un algorithme dont la complexité  $T(n)$ , est donnée par la relation de récurrence ( $n \in \mathbb{R}^{**}$ ):

$$T(1) = 1, \\ T(n) = \alpha T\left(\frac{n}{\beta}\right) + n^\gamma, \quad \text{avec } \alpha > 1, \beta > 1 \text{ et } \gamma > 1.$$

En supposant que  $\gamma = \log_\beta \alpha$ ,

- 1) Calculer  $T(n)$  en résolvant en détail la récurrence.
- 2) Déduire la complexité  $T(n)$  dans le cas ou  $T(1) = 0$ .
- 3) Déterminer  $T(n)$  à l'aide du théorème maître (justifiez votre réponse).

### Exercice 3.

---

Soit la fonction  $F$  (dépendant d'un entier  $n$ ) suivante :

**Fonction F(n)**

```

si n=0 alors
    renvoyer 2 ;
sinon
    renvoyer F(n-1)*F(n-1) ;
fin
    
```

1. Que calcule la fonction  $F(n)$  ?
2. Déterminer la complexité de  $F(n)$  ?
3. Montrer comment améliorer la complexité en proposant un algorithme basé sur le paradigme diviser pour régner ?