Séquence 9 - TD2 - Nouveaux outils théoriques pour les preuves de complexité - Corrigé

Exercice 1 (Quelle stratégie DPR ?).
Pour résoudre un problème donné, vous avez le choix entre trois algorithmes : sur un entrée de taille n :

- `algo1` divise le problème en 5 sous-problèmes de taille $\lceil \frac{n}{2} \rceil$ et combine les solutions en temps $\Theta(n)$
- `algo2` divise le problème en 2 sous-problèmes de taille $n-1$ et combine les solutions en temps $\Theta(1)$
- `algo3` divise le problème en 9 sous-problèmes de taille $\lceil \frac{n}{3} \rceil$ et combine les solutions en temps $\Theta(n^2)$ Quel algorithme choisissez-vous ? Justifier...

Exercice 2 (Résolution de récurrence).
Résoudre la relation de récurrence suivante pour $n = 2^p$:

$$u(n) = 4\,u(\lfloor \frac{n}{2} \rfloor) + n^2$$

Exercice 3 (Complexité moyenne du tri rapide : nombre d'entrées possible infini). Nous allons effectuer une analyse de complexité moyenne du tri rapide. On ne comptera que le nombre de comparaisons pour évaluer la complexité temporelle de l'algorithme.

1. Combien y a-t-il d'entrées différentes possibles pour cet algorithme ?

2. Quelle est la situation la plus défavorable pour l'algorithme de tri rapide ? Quelle est la complexité temporelle dans ce cas ?

3. Quelle semble être la situation la plus favorable ? Quelle est la complexité temporelle dans ce cas ?

4. En dehors de ce cas exceptionnel, quelle est la situation la plus favorable ? Donner un ordre de grandeur asymptotique de la complexité dans ce cas.

5. Nous allons maintenant étudier la complexité moyenne du tri rapide. On se place dans le cas d'un tableau sans doublon, ce qui est pertinent d'un point de vue statistique car la probabilité d'apparition d'un doublon dans un tableau est nulle.

- **a.** En fonction de quelle quantité s'exprime la complexité temporelle de l'algorithme de tripartition ? Combien de comparaisons sont effectuées par cet algorithme ?
- **b.** Pour un tableau de taille n sans doublon, combien y a-t-il de tripartitions possibles ?
- **c.** En vous aidant de la question précédente, exprimez la complexité **moyenne** $T_{\text{moy}}(n)$ de l'algorithme de tri rapide par une formule de récurrence. On fera une moyenne des complexités sur les différentes configurations dénombrées à la question précédente.
- **d.** Résoudre cette récurrence pour obtenir une expression explicite de $T_{\text{moy}}(n)$ en fonction de n.
  Indications :
  - *On commencera par multiplier la relation obtenue pour se débarrasser des dénominateurs*
  - *On procédera ensuite par une technique de combinaison linéaire pour se débarrasser des complexité moyennes $T_{moy}(k)$ intermédiaires.*
  - *On utilisera ensuite une technique de télescopage*
  - *On conclura en se rappelant un équivalent d'une somme « classique »*
- **e.** Donner un équivalent de $T_{\text{moy}}(n)$ lorsque $n \to +\infty$.
- **f.** Conclure, en lien avec la question 2.

Exercice 4 (Études asymptotiques).
Donner un ordre de grandeur asymptotique en Θ si pour, ou en O sinon, pour les complexités suivantes :

1. $T(n) = 5\,T(n-1) + 3n + 2$ et $T(0) = 7$

2. $T(n) = 7\,T(\lfloor \frac{n}{2} \rfloor) + O(n^2)$. (le O($\cdot$) est supposé être une fonction croissante).

3. $T(n) = 2\,T(\lceil \frac{n}{2} \rceil) + n\log_2(n)$