Exercices INF7440 : série #3

1. Dans l'algorithme récursif de tri par fusion, le problème initial de trier un tableau de n valeurs est décomposé en deux sous-problèmes consistant à trier deux sous-tableaux. La taille de chacun de ces sous-tableaux est déterminée telle qu'indiqué dans le pseudo-code suivant (on suppose que le premier index est 1 et qu'on peut affecter ou transmettre en argument des tranches, partielles ou complètes, de tableau) :

```
FONCTION triFusion( int n, keytype S[] ): keytype R[]
DEBUT
  SI n == 1 ALORS
    R <- S;
  SINON
    n1 = n / 2;       // Division entiere.
    n2 = n - n1;

    R1 <- triFusion( n1, S[1..n1] );
    R2 <- triFusion( n2, S[n1+1..n] );
    R  <- fusionner( n1, R1, n2, R2 );
  FIN
  RETOURNER( R );
FIN
```

Supposons alors que l'on remplace la ligne "`n1 = n / 2;`" par la ligne suivante :

```
n1 = 1;
```

a. Quelle est la complexité asymptotique de l'algorithme résultant? De quel type d'analyse s'agit-il (pire cas, tous les cas, meilleur cas, cas moyen)? Y-a-t-il certaines hypothèses que vous devez faire sur `n`? Justifiez brièvement vos réponses.

b. L'algorithme résultant correspond alors à un autre algorithme bien connu. Lequel?

2. (Adapté du manuel) Écrivez une équation de récurrence pour le $n$ième terme de la suite 2, 6, 18, 54, ... Trouvez ensuite une formule explicite, puis utilisez l'induction mathématique pour prouver que votre formule explicite satisfait bien l'équation de récurrence.

3. En utilisant le théorème général, trouvez les solutions pour l'équation de récurrence suivante, en fonction de a ($a \in \mathcal{N}, a \geq 1$):

- $T(1) = 1$
- $T(n) = aT(n/2) + n^2$

4. Trouvez les diverses solutions de l'équation de récurrence suivante en fonction de k (pour k un nombre réel non-négatif):

$$T(1) = 1$$

$$T(n) = 4T(n/2) + n^k$$

5. Soit un algorithme récursif diviser-pour-régner qui décompose un problème de taille n en trois (3) sous-problèmes de taille $n-2$. Donnez les équations de récurrence qui donnent une *borne inférieure* de la complexité asymptotique de cet algorithme et trouvez la solution de ces équations. Comme toujours, vous pouvez supposer que n est d'une forme appropriée.

Exercices tirés du manuel.

Exercices de l'annexe B (pp. 582–587)

- 1. (a), (b) et (d)
- 20. (a), (b) et (d)
- 24.

  Note : L'une des questions indique "$nT(n) = \ldots$". Le n devant $T(n)$ est une erreur et ne devrait pas apparaître.
- 25.

  Note : Même remarque que pour 24.