Algorithmique : exercices de complexité

Julien Vion

1. Au cours de l'année scolaire, ma première note était de ½₀. Ensuite, à chaque examen, j'ai amélioré mon résultat précédent de 1 point. Quelle est la moyenne de mes 15 premières notes ?

2. Déterminez la complexité temporelle, en notation Landau, de chacun des algorithmes suivants (n est un paramètre de l'algorithme) :

   (a)
   ```
   for (int i = 0; i < 10; i++) {
     for (int j = 0; j < n; j++) {
       System.out.println("Hello World");
     }
   }
   ```

   (b)
   ```
   for (int i = 0; i < n; i++) {
     for (int j = 0; j < 2*n + 1; j++) {
       System.out.println("Hello World");
     }
   }
   ```

   (c)
   ```
   for (int i = 0; i < n; i++) {
     for (int j = 0; j < n; j++) {
       System.out.println("Hello World");
     }
   }
   for (int k = 0; k < n; k++) {
     System.out.println("Hello World");
   }
   ```

   (d)
   ```
   for (int i = 0; i < n; i++) {
     for (int j = i; j < n; j++) {
       System.out.println("Hello World");
     }
   }
   ```

   (e)
   ```
   for (int i = 0; i < n; i++) {
     int t = 1;
     while (t < n) {
       System.out.println("Hello World");
       t *= 2;
     }
   }
   ```

3. Soient trois fonctions (méthodes statiques) $A(n)$, $B(n)$ et $C(n)$ dont les temps de calcul sont respectivement $T_A(n)$, $T_B(n)$ et $T_C(n)$, et le code suivant :

```
A(n);
if (n < 1000) {
    B(n);
} else {
    for (int i = 0; i < n; i++) {
      C(n);
    }
}
```

Quelle est la complexité du code précédent, dans les cas de figure suivants :

(a) $T_A(n) \in \Theta(n)$, $T_B(n) \in \Theta(n^2)$, et $T_C(n) \in \Theta(\log n)$.

(b) $T_A(n) \in \Theta(n^2)$, $T_B(n) \in \Theta(n^2)$, et $T_C(n) \in \Theta(\log n)$.

(c) $T_A(n) \in \Theta(n^2)$, $T_B(n) \in \Theta(n^3)$, et $T_C(n) \in \Theta(\log n)$.

4. Écrivez un algorithme qui renvoie un tableau contenant les m premiers éléments d'un tableau de n éléments. On suppose $m \leq n$. Quel est la complexité de votre algorithme ?

   On fait un banc d'essai en extrayant les 100 premiers éléments d'un tableau de 10 000 éléments aléatoires. L'algorithme met 30 ms à répondre. Combien de temps mettrait-il à répondre sur un tableau d'un million d'éléments ? Et si on veut extraire les 1 000 premiers éléments d'un tableau de 10 000, un million d'éléments ?

5. Écrivez un algorithme qui recherche le plus grand élément d'un tableau. L'algorithme doit renvoyer *l'indice* de l'élément le plus grand. Quel est la complexité de votre algorithme ?

   Si on faisait un banc d'essai sur un tableau de 10 000 éléments aléatoires et que votre algorithme mettait 30 ms à répondre, combien de temps mettrait-il à répondre sur un tableau d'un million d'éléments ?

6. On veut maintenant rechercher les m plus grands éléments d'un tableau non trié de taille n. Une idée serait de rechercher le plus grand élément en utilisant l'algorithme de la question 4, le stocker dans une nouvelle liste, de supprimer l'élément, et recommencer l'opération m fois. On peut supposer que la suppression d'un élément d'un tableau se fait en $\Theta(n)$.

   Quelle est la complexité globale de l'opération ?

   Si on trie le tableau, peut-on trouver une autre méthode pour retrouver les m plus grands éléments ? Quelle complexité obtient-on ? Peut-on exploiter cette méthode sans toucher au tableau initial ?

   Essayez d'implémenter en Java et évaluer les temps d'exécution de ces différentes méthodes, par exemple pour $n = 10^5$, $n = 10^6$, $m = 10^2$, $m = 10^3$. N'hésitez pas à utiliser une ArrayList plutôt qu'un tableau : vous pourrez alors utiliser les méthodes List#remove, Collections.sort ou List#sort, et le constructeur par recopie déjà définis dans l'API.