TD Invariants n°2

Quelques invariants

Ce TD comporte plusieurs exercices indépendants. L'exercice I est très simple. Les exercices II et III sont plus complexes. L'exercice IV est bien plus complexe.

I. Redécouvrir l'exponentiation rapide.

L'objectif de cet exercice est de calculer efficacement a^n, pour $a \in \mathbb{Q}$ et $n \in \mathbb{N}$. On supposera que la multiplication de deux rationnels est une opération élémentaire.

Q1. On suppose n pair. Supposons pouvoir calculer $a^{n/2}$ avec un algorithme $\mathscr{A}_{n/2}$. Construire un algorithme $\mathscr{A}_n$ calculant efficacement a^n. On utilisera habilement $\mathscr{A}_{n/2}$.

Q2. Soit n une puissance de deux. Trouver un algorithme calculant efficacement a^n.

Q3. Avec un O, déterminer le nombre d'opérations élémentaires de l'algorithme trouvé en Q2. On supposera le calcul de $n/2$ à partir de n possible en une opération élémentaire.

Q4. Soit $\textsc{Pow}(a, n)$ l'algorithme de la question Q2. Prouver sa correction par récurrence, pour tout entier $n \in \mathbb{N}$, qui est une puissance de deux.

Q5. Construire un algorithme calculant a^n pour $n \in \mathbb{N}$ qui n'est pas *forcément* une puissance de deux.

II. Tri en une seule boucle ?

Considérons l'algorithme ci-contre. La tableau A commence à 0.

Entrée. Un tableau A de taille n.
Sortie. Le tableau A modifié.

```
i ← 0
Tant que i < n faire
    Si i = 0 ou A[i] ≥ A[i − 1] alors
        i ← i + 1
    Sinon
        Échanger A[i] et A[i − 1]
        i ← i − 1
    Fin si
Fin tant que
```

Q1. Donner les étapes de l'algorithme sur le tableau A :

$$A = [10, 20, 30, 40, 50, 25]$$

jusqu'à la fin de l'algorithme. On s'intéressera particulièrement au contenu de A et la valeur de i après chaque itération de la boucle **Tant que**.

Q2. Expliquer pourquoi l'algorithme trie n'importe quel tableau A. Introduire un invariant $(\mathscr{I})$. Le démontrer. Conclure sur la correction de l'algorithme.

Q3. Démontrer que l'algorithme termine à l'aide d'un variant. Donner un majorant raisonnable du temps d'exécution à l'aide d'un O.

III. Tri bulle.

On considère l'algorithme ci-après, que l'on nommera $\textsc{TriBulle}(A)$. On supposera que le tableau A commence à 1.

Q1. Trouver un invariant $\mathscr{I}(i)$ qui est vrai à la fin de la boucle **Pour** extérieure.

Q2. Démontrer que l'invariant $\mathscr{I}(i)$ est bien vérifié, quel que soit i.

Q3. En déduire sur la correction du tri bulle.

> **Indication.** Afin de visualiser comment fonction l'algorithme TriBulle, on pourra essayer un exemple.
>
> Si vous ne voyez pas l'idée de l'algorithme, vous pourrez regarder cette animation.

Pour $i \in [\![1, n]\!]$ **faire**
  Pour $j \in [\![1, n-i]\!]$ **faire**
    Si $A[j] > A[j+1]$ **faire**
      Échanger $A[j]$ et $A[j+1]$
    Fin si
  Fin pour
Fin pour

IV. Un problème de graphes.

On considère un graphe pondéré orienté acyclique $G = (V, E, w)$, où $V = [\![1, n]\!]$. L'objectif de cet exercice est de trouver la longueur du plus long chemin dans G.

On fixe top un *tri topologique* de V. Un *tri topologique* de V est une permutation de V telle que, pour tout couple d'entiers $(i, j) \in [\![1, n]\!]^2$, si $(\mathrm{top}_i, \mathrm{top}_j) \in E$, alors $i \leq j$.

On considère l'algorithme ci-contre.

Dans cet algorithme, on supposera que le maximum sur l'ensemble vide est 0.

$(L_1, \dots, L_n) \leftarrow \underbrace{(0, \dots, 0)}_{n}$
Pour $i \in [\![1, n]\!]$ **faire**
  $v \leftarrow \mathrm{top}_i$
  $L_v \leftarrow \max\{L_u + w(u, v) \mid u \in V\}$
Fin pour
Renvoyer $\max_{i \in [\![1,n]\!]} L_i$

On supposera que $w(u, v)$ est nul si $(u, v) \notin E$.

Q1. Montrer que l'algorithme vérifie l'invariant $\mathscr{I}(i)$: après i itérations de la boucle **Pour**, l'élément L_{top_i} contient la longueur du plus long chemin qui termine par top_i.

Q2. Quelle est la complexité de l'algorithme ? On exprimera la réponse sous la forme d'un Θ en terme de V et E. Justifier votre réponse.

Ces exercices sont extraits des cours d'informatique « Algorithmes et structures de données » de l'École polytechnique fédérale de Zurich.