EPFL - Information, Calcul et Communication - Partie théorique

Exercices additionnels

Les exercices ci-dessous sont optionnels et ont pour but de vous donner de quoi vous entraîner à écrire (et lire) des algorithmes. Les solutions ne seront pas publiées, mais vous êtes les bienvenu.e.s d'échanger des idées pour trouver la solution à chacun de ces problèmes en utilisant le forum Ed Discussion.

Cinq algorithmes à écrire

Après avoir résolu chaque problème ci-dessous, vous êtes invités à réfléchir si la solution que vous avez trouvée est la plus efficace en temps de calcul. La meilleure (i.e., la plus petite) complexité temporelle qu'il est possible d'obtenir pour chaque algorithme est donnée au bas de la page.

a) Soit L une liste de n nombre entiers. Ecrire un algorithme qui trouve la plus *grande* différence en valeur absolue entre deux nombres de la liste.

Exemple: Si $L = (-3, 17, 15, -22, 3)$ et $n = 5$, alors la sortie doit être $39 = |17 - (-22)|$.

b) Soit L une liste de n nombre entiers. Ecrire un algorithme qui trouve la plus *petite* différence entre deux nombres de la liste.

Exemple: Si $L = (-3, 17, 15, -22, 3)$ et $n = 5$, alors la sortie doit être $2 = |17 - 15|$.

c) Supposons qu'on dispose d'un algorithme **appartient à**(L, k) dont la sortie soit oui si la taille de la liste L est plus grande ou égale à k (i.e., si $L(k)$ est un élément appartenant à la liste L), et non dans le cas contraire. Ecrire un autre algorithme utilisant celui-ci qui calcule la taille de la liste L.

Exemple: Si $L = (-3, 17, 15, -22, 3)$, alors la sortie doit être simplement $n = 5$.

NB: On suppose ici que la complexité temporelle de l'algorithme **appartient à**(L, k) est $\Theta(1)$.

d) Supposons qu'on dispose d'un algorithme **aléatoire**(n) qui tire un nombre entier uniformément au hasard entre 1 et n. Ecrire un autre algorithme utilisant celui-ci qui à partir d'une liste ordonnée L de n nombres entiers, génère une liste aléatoire A qui soit une version "dans le désordre" de la liste L (donc tous les nombres de L doivent se retrouver dans la liste A).

Exemple: Si $L = (-22, -3, 3, 15, 17)$ et $n = 5$, alors la sortie peut être par exemple $A = (15, 3, 17, -22, -3)$.

NB: On suppose ici que la complexité temporelle de l'algorithme **aléatoire**(n) est $\Theta(n)$.

e) Soit n un nombre entier positif et A un tableau de dimensions $2 \times n$ rempli de nombres entiers positifs tels que *dans chacune des 2 lignes du tableau, tous les nombres sont différents.* Par exemple, si $n = 7$, un tel tableau pourrait être:

$$A = \begin{pmatrix} 4 & 3 & 11 & 2 & 12 & 1 & 5 \\ 5 & 2 & 12 & 13 & 15 & 4 & 3 \end{pmatrix}$$

Ecrire un algorithme dont les entrées soient un tableau A de ce type et n le nombre de colonnes de A, et dont la sortie soit oui s'il existe un numéro de colonne $i_0 \in \{1, \ldots, n\}$ tel que

$$A(1, i_0) \geq A(1, i) \quad \text{et} \quad A(2, i_0) \geq A(2, i) \quad \text{pour tout } i \in \{1, \ldots, n\}$$

et non dans le cas contraire (dans l'exemple ci-dessus, la sortie doit donc être oui).

a) $\Theta(n)$ b) $\Theta(n \cdot \log_2(n))$ c) $\Theta(\log_2(n))$ d) $\Theta(n^2)$ e) $\Theta(n)$

Exercice. On considère les deux algorithmes suivants:

machin

entrée : *nombre entier positif* n
sortie : *???*

Si $n = 1$
  Sortir: 1

Sortir : $2^{(\textbf{bidule}(n-1)+1)}$

bidule

entrée : *nombre entier positif* n
sortie : *???*

Si $n = 1$
  Sortir: 0

Sortir : $\log_2(\textbf{machin}(n-1))$

a) Quelle est la sortie de l'algorithme **bidule** lorsque $n = 4$ en entrée?

b) Quelle est la complexité temporelle de l'algorithme **bidule**? (utiliser la notation $\Theta(\cdot)$)

c) Réécrire l'algorithme **machin** de façon récursive, mais sans faire appel à **bidule**.

d) En déduire quelle est la sortie de **machin** pour une valeur quelconque de $n \geq 1$ en entrée.