L2 Informatique - Algorithmique - Session 1

mardi 16 décembre 2014. 09h00-11h00. A 400.

La précision et la clarté de votre rédaction sont *fondamentales*. Documents interdits. Durée 2h00. Le barème indiqué est *approximatif*.

Exercice 1. [2pts] Quelle est la valeur de la somme des entiers compris entre 39 et 78 ? Soit n un entier strictement positif, calculez la somme

$$\sum_{i=0}^{n} 3^i.$$

Exercice 2. [4pts] Dessinez l'arbre binaire de décision du tri par sélection par le minimum pour les instances de taille 3. Les branches droites correspondent aux comparaisons qui sont satisfaites, les feuilles doivent contenir la liste avant le tri. De manière plus générale, combien de feuilles y-a-t-il au minimum dans l'arbre de décision associé à un algorithme de tri pour une instance de taille n ?

Exercice 3. [2pts] Écrivez un algorithme `MinMax(L):entier` qui renvoie le *premier* indice du terme le plus petit de la liste L ainsi que le *dernier* indice du terme le plus grand de la liste L. Par exemple pour la liste $[2, 1, 4, 3, 7, 9, 2, 1, 9, 3]$, l'algorithme renvoie les valeurs 2 et 9 (on suppose ici que l'indexation commence par 1). On veut que l'algorithme trouve ces deux indices *en une seule passe* sur la liste. Quelle est la complexité de cet algorithme ?

Exercice 4. [2.5pts] Écrivez l'expression arithmétique suivante sous forme postfixée :

$$(2 + 3) \times (7 - 1) + 2$$

Évaluez cette expression à l'aide d'une pile. On rappelle qu'on lit les termes de l'expression postfixée et que, selon qu'il s'agisse d'un opérande x ou d'un opérateur $\star$, on empile x ou on dépile les deux valeurs a et b au sommet de la pile et on empile $a \star b$.

Faites un tableau contenant les différents états de la pile au fur et à mesure de l'évaluation de l'expression postfixée que vous aurez trouvée. Quel est le résultat final dans la pile ?

Exercice 5. [6pts] Soient u et v deux séquences de valeurs dans un ensemble E de longueurs respectives n et m avec $n \leq m$. On dit que u est une *sous-séquence* de v s'il existe une application strictement croissante $\sigma : [1; n] \to [1; m]$ telle que $\forall i \in [1; n],\ u_i = v_{\sigma(i)}$. Par exemple le mot "cas" est une sous-séquence du mot "carnages" (ou encore "carnages"), ainsi que le mot "rage" ("carnages"). La séquence vide est toujours considérée comme une sous-séquence de n'importe quelle séquence.

En supposant que la séquence v est constituée de n valeurs distinctes, combien peut-on construire de sous-séquences u de v ?

Écrivez un algorithme `EstSousSeq(u,v):booleen` qui renvoie `vrai` si u est une sous-séquence de v et `faux` sinon. On notera $u[i]$ le i-ème terme de la séquence u. Estimez la complexité de cet algorithme dans le meilleur des cas et dans le pire des cas en fonction des longueurs n et m des séquences u et v.

Exercice 6. [6pts] Écrivez un algorithme `Dichotomie(L,x):entier` qui recherche la valeur x dans une liste L supposée triée dans l'ordre croissant avec la méthode de la dichotomie. On supposera que la liste est indexée à partir de 1 et que $\sharp L$ désigne le nombre de termes de la liste. L'algorithme devra renvoyer le premier indice de x où x apparaît dans la liste L ou 0 si x n'apparaît pas dans la liste L.

Estimez la complexité de votre algorithme.

Rappel important ! pour toutes les questions de complexité, précisez :

(1) ce que désigne la/les variable(s) de la fonction de complexité T ;

(2) quelles instructions vous comptabilisez et pourquoi vous les considérez représentatives de l'algorithme ;

(3) s'il y a lieu de distinguer différentes complexités, meilleur des cas, pire des cas (le cas échéant, le complexité moyenne n'est pas demandée).