Numéro d'anonymat :

Examen 2I003

Mercredi 4 Janvier 2017, 2 heures
aucun document autorisé

Exercice 1 – Le pic - 10 points

On dit qu'un tableau `T[0..n-1]` de n entiers admet un *pic* s'il existe un indice p (compris au sens large entre 0 et $n-1$) tel que :
– le sous-tableau `T[0..p]` est strictement croissant
– le sous-tableau `T[p..n-1]` est strictement décroissant.
Par exemple :
– `[2,3,5,6,8,9,7,1]` admet un pic en position 5,
– `[45,13,6,4,2]` admet un pic en position 0,
– `[1,3,5,7,23]` admet un pic en position 4.
Dans cet exercice on considère des tableaux de n entiers qui admettent un pic et on étudie des algorithmes qui calculent la position de ce pic.

Le pic en itératif

On dispose de la fonction `Lambda(T, k)` ainsi définie, pour $0 \leq k \leq n-1$:

```
def Lambda(T, k):
    n = len(T)
    if k == n - 1:
        return True
    return T[k] > T[k + 1]
```

On considère la fonction `picIte(T)` ainsi définie :

```
def picIte(T):
    k = 0
    while not(Lambda(T, k)):
        k = k + 1
    return k
```

Question 1

Donner le résultat de `picRec(ExT,0,13)`, avec `ExT = [1,3,5,7,8,13,16,17,19,23,45,11,6,4]`.

Question 2

Montrer que `picIte(T)` se termine.

Question 3
On note k_i la valeur de k à la fin de la i-ème itération (k_0 vaut 0). Montrer par récurrence sur i que, à la fin de la i-ème itération, $k_i = i$ et `T[0..i]` est strictement croissant.

Question 4
Déduire des deux questions précédentes que `picIte(T)` calcule la position du pic de T.

Question 5
Quelle est la complexité (en nombre de comparaisons entre éléments du tableau) de `picIte(T)` dans le meilleur cas ? Dans le pire cas ?

Le pic en récursif

On considère la fonction `picRec(T,i,j)` ainsi définie, pour $0 \leq i \leq j \leq n-1$:

```
def picRec(T, i, j):
    print("Recherche␣du␣pic␣entre␣", i, "␣et␣", j)
    if i == j:
        return i
    m = (i + j) // 2
    print("m␣=␣", m)
    if T[m] < T[m + 1]:
        return picRec(T, m + 1, j)
    return picRec(T, i, m)
```

On rappelle que `//` est l'opérateur de la division entière (par exemple : `7//2 = 3, 8//2 = 4`).

Question 6
Exécuter l'appel de `picRec(ExT,0,13)`, avec `ExT = [1,3,5,7,8,13,16,17,19,23,45,11,6,4]`, en donnant les affichages successifs et le résultat final.

Question 7
Soit i, j des entiers naturels tels que $i < j$. On pose $m = (i+j) \,//\, 2$ et $t = j - i + 1$. Montrer que :
a) $i \leq m < j$,
b) $1 \leq m - i + 1 < t$ et $1 \leq j - m < t$.

Question 8
Montrer, par récurrence forte sur $t = j - i + 1$, que `picRec(T,i,j)` se termine, pour $t \leq 1$.

Question 9
Montrer, par récurrence forte sur $t = j - i + 1$, que `picRec(T,i,j)` calcule la position du pic de `T[i..j]`.

Question 10
On considère la fonction `pic(T)` ainsi définie :

```
def pic(T):
    return picRec(T, 0, len(T)-1)
```

Déduire des deux questions précédentes que `pic(T)` calcule la position du pic de T.

Question 11
Quelle est la complexité (en nombre de comparaisons entre éléments du tableau) de `pic(T)` ? Justifier la réponse.

Exercice 2 – Graphes et arbres - 10 points

Dans cet exercice, $G = (V, E)$ désigne un graphe non orienté. $n = |V|$ désigne le nombre de sommets, et $m = |E|$ le nombre d'arêtes.

Question 1
Dans cette question, on suppose que $G = (V, E)$ est défini par $V = \{1, 2, 3, 4, 5\}$ et $E = \{\{1, 2\}, \{1, 3\}, \{1, 5\}, \{3, 4\}, \{2, 4\}\}$. Que valent m et n ? Représentez la matrice sommets-sommets M de G.

Question 2

1. Qu'est ce qu'un graphe connexe ? Est-ce que le graphe de la question précédente est connexe ?
2. Qu'est ce qu'un arbre ? Est-ce que le graphe de la question précédente est un arbre ? Quelles sont les arêtes que l'on peut rajouter ou supprimer pour obtenir un arbre ?

Question 3

Soit la relation $\mathcal{R}$ définie sur V^2 par $u\mathcal{R}v$ ssi $u = v$ ou si il existe une arête entre u et v.

1. Donnez la définition de $\mathcal{R}$ pour le graphe de la question 1.
2. Décrire le principe d'un algorithme qui permet de calculer la fermeture transitive $\mathcal{R}'$ de $\mathcal{R}$ dans le cas général. Ne pas écrire l'algorithme en (pseudo)-code. Quelle est sa complexité ?
3. Que vérifie $\mathcal{R}'$ si G est connexe ? En déduire un algorithme qui teste si un graphe est connexe. Quelle est sa complexité ?

Question 4
On rappelle qu'un graphe non orienté $G = (V, E)$ est minimal connexe si il est connexe et que, pour tout arête $e \in E$, $G' = (V, E - \{e\})$ ne l'est pas.

1. Représentez un graphe de 5 sommets qui est connexe, mais pas minimal connexe.

2. Démontrez que, si G est un arbre, alors G est minimal connexe. Utilisez la contraposée.

3. Démontrez que si G est minimal connexe, alors G est un arbre. Utilisez la contraposée.

Question 5

1. Rappelez sans démonstration la relation entre le nombre de sommets et le nombre d'arêtes d'un graphe G connexe.
2. Démontrez que si $m = n - 1$ et que G est connexe, alors G est un arbre. La démonstration doit être effectuée par l'absurde.
3. Que pensez-vous de la réciproque (sans démonstration).
4. En déduire un algorithme simple qui teste si un graphe G est un arbre.