LU2IN003 Examen Algorithmique
Aucun document autorisé

Vendredi 19 mai 2023 - Durée 2 heures

Exercice 1 : Calcul du saut (10 points)
Dans cet exercice, on considère un tableau tab de n entiers non vide ($n > 0$). Un **saut** est un couple d'entiers (i, j) avec $0 \leq i \leq j < n$ et la **valeur** associée est $tab[j] - tab[i]$. On à calculer pour un tableau tab non vide, la valeur maximale $\mathcal{S}(tab)$ d'un saut.

1. On note dans cette question $tab_1 = (3, 5, 1, 2, 8, 3)$.
   (a) Quelle sont les valeurs des sauts $(2, 5)$ et $(1, 3)$ pour le tableau tab_1 ? Justifiez votre réponse ;
   (b) Quelle est la valeur $\mathcal{S}(tab_1)$? Justifiez votre réponse.

> **Solution:**
> (a) La valeur du saut $(2, 5)$ vaut $t[5] - t[2] = 3 - 1 = 2$; la valeur du saut $(1, 3)$ vaut $t[3] - t[1] = 2 - 5 = -3$;
> (b) La valeur maximale d'un saut est obtenue pour le couple $(i, j) = (2, 4)$. En effet, $tab_1[4] = 8$ est le plus grand élément de tab_1 et $tab_1[2] = 1$ le plus petit. On a donc bien que $\mathcal{S}(tab) = t[4] - t[2] = 8 - 1 = 7$.

2. Soient max et min les valeurs maximales et minimales stockées dans le tableau tab.
   (a) Démontrez que $\mathcal{S}(tab) \geq 0$;
   (b) Démontrez que $\mathcal{S}(tab) \leq max - min$;
   (c) Est-ce que l'égalité $\mathcal{S}(tab) = max - min$ est toujours vérifiée ? Justifiez votre réponse.

> **Solution:**
> (a) tab est non vide. Le saut $(0, 0)$ est de valeur 0. On en déduit que $\mathcal{S}(tab) \geq 0$;
> (b) Pour tout saut (i, j), $tab[j] \leq max$ et $tab[i] \geq min$. Donc, si on suppose que $(i^\star, j^\star)$ est un saut de valeur maximale, $\mathcal{S}(tab) = t[j^\star] - t[i^\star] \leq max - min$;
> (c) Cette propriété est fausse dans le cas général. Par exemple, si $tab = (3, 1)$, $max = 3$, $min = 1$ et $\mathcal{S}(tab) = 0$.

3. On considère la suite d'ensembles de couples d'entiers définie par $S_{-1} = \{(0, 0)\}$ et pour tout $i \in \{0, \ldots, n-1\}$, $S_i = \{(i, j), j \in \{i, \ldots, n-1\}\}$.
   (a) Donnez les ensembles S_i pour un tableau de taille $n = 6$ et $i \in \{0, \ldots, 5\}$;
   (b) Démontrez que dans le cas général, $\bigcup_{i=0}^{n-1} S_i$ est l'ensemble de tous les sauts d'un tableau tab de taille n.

Solution:

(a) $S_0 = \{(0,0,(0,1),(0,2),(0,3),(0,4),(0,5)\}$, $S_1 = \{(1,1),(1,2),(1,3),(1,4),(1,5)\}$, $S_2 = \{(2,2),(2,3),(2,4),(2,5)\}$, $S_3 = \{(3,3),(3,4),(3,5)\}$, $S_4 = \{(4,4),(4,5)\}$ et $S_5 = \{(5,5)\}$;

(b) On observe que pour tout $i \in \{0,\dots,n-1\}$, S_i est l'ensemble de tous les sauts de *tab* dont le premier élément est i. Donc, $\bigcup_{i=0}^{n-1} S_i$ est l'ensemble de tous les sauts d'un tableau de taille n.

4. On considère la fonction itérative `indiceMax(tab)` pour *tab* un tableau de n entiers donné par l'Algorithme 1. L'instruction `for j in range(1,len(tab))` consiste à prendre les valeurs de j dans l'ordre $(1,2,\dots,n-1)$.

```
def indiceMax(tab):
        jstar=0
        for j in range(1,len(tab)):
                if tab[j]>tab[jstar]:
                        jstar=j
        return jstar
```

Algorithme 1 : Définition de la fonction itérative `indiceMax`.

(a) Démontrez que `indiceMax(tab)` se termine ;

(b) Que retourne cette fonction ? Exprimez un invariant de boucle.

(c) Quel est le nombre de comparaisons effectuées par l'appel `indiceMax(tab)` pour un tableau de taille n. Justifiez votre réponse.

(d) En déduire la complexité de cette fonction. Justifiez votre réponse.

Solution:

(a) Le corps de boucle est constitué d'un test et d'une affection. Chacune de ses exécutions se termine. Il est effectué $n-1$ fois, donc il se termine ;

(b) Cette fonction retourne le plus petit indice $jstar \in \{0,\dots,n-1\}$ tel que $tab[jstar]$ est un éléments de t de valeur maximale ;

Pour tout $j^\star \in \{0,\dots,n-1\}$, on considère que $jstar_{j^\star}$ est la valeur de $jstar$ à la fin de l'itération $j = j^\star$. L'invariant s'exprime alors par : $jstar_{j^\star}$ est la plus petite valeur de l'ensemble $\{0,\dots,j^\star\}$ telle que $tab[jstar_{j^\star}]$ est la valeur maximale du sous-tableau $tab[0\dots j^\star]$;

(c) Dans tous les cas, `indiceMax(tab)` parcourt le tableau et compare l'élément courant $t[i]$ au maximum du sous-tableau $tab[0\dots i-1]$. Il effectue donc $n-1$ comparaisons ;

(d) Il y a exactement une comparaison par itération, et l'exécution du corps de boucle est en $\Theta(1)$. La complexité est donc proportionnelle au nombre de comparaisons, soit en $\Theta(n)$.

On considère maintenant la fonction itérative `sautMax` définie par l'Algorithme 2. On rappelle que *tab* est un tableau de n entiers non vide ($n \geq 1$).

L'instruction `for i in range(0,len(tab))` consiste à prendre les valeurs de i dans l'ordre $(0, 1, \ldots, n-1)$. L'instruction `(resi,resj)=(i,jmax)` affecte la valeur de i à $resi$ et celle de $jmax$ à $resj$. La fonction retourne les deux valeurs $resi$ et $resj$.

Enfin, pour deux entiers i et j tels que $0 \leq i \leq j \leq n$, la notation Python `tab[i:j]` désigne le sous-tableau $tab[i \ldots j-1]$ de tab. Par extension, `tab[i:]` correspond à $tab[i \ldots n-1]$ et `tab[:j]` à $tab[0 \ldots j-1]$.

```
def sautMax(tab):
    n=len(tab)
    resi=0;resj=0
    for i in range(0,len(tab)):
        jmax=i+indiceMax(tab[i:])
        if (tab[jmax]-tab[i])>(tab[resj]-tab[resi]):
            (resi,resj)=(i,jmax)
    return (resi,resj)
```

Algorithme 2 : Définition de la fonction itérative `sautMax`.

5. Démontrez la terminaison de la fonction `sautMax`.

Solution: Par hypothèse, tous les exécutions de `indiceMax` se terminent. Les autres instructions de la boucle sont élémentaires (un test et une affectation). La boucle est de plus exécutée n fois. Donc, la fonction `sautMax` se termine.

6. On définie les suites d'entiers suivantes :
   — Pour tout $i^\star \in \{-1, 0, \ldots, n-1\}$, $resi_{i^\star}$ et $resj_{i^\star}$ vérifient $resi_{-1} = resj_{-1} = 0$ et pour $i^\star \geq 0$, $resi_{i^\star}$ et $resj_{i^\star}$ sont les valeurs respectives des variables $resi$ et $resj$ à la fin de l'itération $i = i^\star$;
   — Pour tout $i^\star \in \{0, \ldots n-1\}$, $jmaxx_{i^\star}$ désigne la valeur de la variable $jmax$ à la fin de l'itération $i = i^\star$.
   (a) Donnez les valeurs des suites $resi_{i^\star}$, $resj_{i^\star}$ et $jmax_{i^\star}$ pour $tab_2 = (4, 3, 5, 7, 1, 6)$.
   (b) Démontrez dans le cas général que, pour tout $i^\star \in \{0, \ldots, n-1\}$, le couple $(i^\star, jmax_{i^\star}) \in S_{i^\star}$ et est un saut de valeur maximale dans $S_{i^\star}$ (la suite des ensembles $S_{i^\star}$ a été définie dans la question 3).

Solution:

(a)

$i^\star$	-1	0	1	2	3	4	5
$resi_{i^\star}$	0	0	1	1	1	4	4
$resj_{i^\star}$	0	3	3	3	3	5	5
$jmax_{i^\star}$	$\star$	3	3	3	3	5	5

(b) On se place au démarrage de l'itération $i = i^\star$. Le sous-tableau `tab[i:]` correspond à $tab[i^\star \ldots n-1]$ et est de taille $n-1-i^\star+1 = n-i^\star$. On en déduit que la valeur de retour de `indiceMax` est dans l'ensemble $\{0, \ldots, n-i^\star-1\}$ et donc $jmax_{i^\star} \in \{i^\star, \ldots, n-1\}$.

Ainsi, $(i^\star, jmax_{i^\star})$ est bien un saut dont le premier élément est $i^\star$, et donc apartient $S_{i^\star}$.

De plus, $t[jmax_{i^\star}]$ est le plus grand élément de $tab[i^\star \dots n-1]$, il maximise donc la valeur $t[jmax_{i^\star}] - t[i^\star]$ et ainsi le saut $(i^\star, jmax_{i^\star})$ est de valeur maximale dans $S_{i^\star}$.

7. Pour tout $i^\star \in \{-1, 0, \dots, n-1\}$, on considère l'invariant de boucle $\mathcal{P}(i^\star)$: le couple $(resi_{i^\star}, resj_{i^\star})$ est un saut de valeur maximale dans $\bigcup_{i=-1}^{i^\star} S_i$.
   (a) Démontrez par récurrence que pour tout $i^\star \in \{-1, 0, \dots n-1\}$, $\mathcal{P}(i^\star)$ est vérifiée ;
   (b) En déduire la validité de la fonction `sautMax`.

Solution:

(a) Cette propriété se démontre par récurrence faible sur $i^\star$.

Base Pour $i^\star = -1$, au démarrage de la boucle, $resi_{-1} = resj_{-1} = 0$ et $S_{-1} = \{(0, 0)\}$. $\mathcal{P}(-1)$ est donc vérifiée.

Induction Soit $i^\star \in \{-1, \dots, n-2\}$, supposons que $\mathcal{P}(i^\star)$ est vérifiée. On montre que $\mathcal{P}(i^\star + 1)$ est vérifiée.

On se place au démarrage de l'itération $i = i^\star + 1$. D'après la question 6, le saut $(i^\star + 1, jmax_{i^\star+1}) \in S_{i^\star+1}$ et est de valeur maximale dans $S_{i^\star+1}$. D'autre part, par hypothèse de récurrence, $(resi_{i^\star}, resj_{i^\star})$ est un saut de valeur maximale dans $\bigcup_{i=-1}^{i^\star} S_i$. On a deux cas à considérer :

1. Si le test est vérifié, la valeur du saut $(i^\star+1, jmax_{i^\star+1})$ est supérieure à celle du saut $(resi_{i^\star}, resj_{i^\star})$. Donc, la valeur du saut $(i^\star+1, jmax_{i^\star+1})$ est supérieure à tous les sauts de $\bigcup_{i=-1}^{i^\star} S_i$. On en déduit que $(i^\star + 1, jmax_{i^\star+1})$ est un saut de valeur maximale dans $\bigcup_{i=-1}^{i^\star+1} S_i$;
2. Sinon, la valeur du saut $(resi_{i^\star}, resj_{i^\star})$ est supérieure à celle de $(i^\star+1, jmax_{i^\star+1})$. On en déduit que la valeur du saut $(resi_{i^\star}, resj_{i^\star})$ est supérieure à celle de tous les éléments de $S_{i^\star+1}$ et donc $(resi_{i^\star}, resj_{i^\star})$ est un saut de valeur maximale dans $\bigcup_{i=-1}^{i^\star+1} S_i$.

Conclusion La proposition $\mathcal{P}(i^\star)$ est donc vérifiée par récurrence faible pour toute valeur $i^\star \in \{-1, \dots, n-1\}$.

(b) En sortie de boucle, $i^\star = n-1$ et $\mathcal{P}(n-1)$ est vérifiée. Le couple de variables $(resi, resj) = (resi_{n-1}, resj_{n-1})$ est un saut de valeur maximale dans $\bigcup_{i=-1}^{n-1} S_i$. Or, d'après la question 3, $\bigcup_{i=-1}^{n-1} S_i = \bigcup_{i=0}^{n-1} S_i$ correspond à l'ensemble de tous les sauts associés à *tab*. On en déduit que le saut renvoyé par `sautMax` est de valeur maximale.

8. On souhaite maintenant évaluer la complexité de la fonction `sautMax`.
   (a) Pour $i^\star \in \{0, \dots, n-1\}$, quel est le nombre de comparaisons $c_{i^\star}$ effectuées par l'itération $i = i^\star$? Justifiez votre réponse.
   (b) Quel est le nombre de comparaisons totales effectuées ? Justifiez votre réponse.
   (c) En déduire la complexité de la fonction `sautMax`.

Solution:

(a) Pour $i^\star \in \{0, \ldots, n-1\}$ et $i = i^\star$, le tableau `tab[i:]` correspond à $tab[i^\star \ldots n-1]$ et a donc pour taille $n - 1 - i^\star + 1 = n - i^\star$. L'appel de la fonction `indiceMax` effectue donc $n - i^\star - 1$ comparaisons. On a en tout $c_{i^\star} = n - i^\star$ comparaisons à l'itération $i = i^\star$;

(b) Le nombre total de comparaisons est donc $\sum_{i=0}^{n-1} c_i = \sum_{i=0}^{n-1}(n-i) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$;

(c) Le nombre d'instructions est du même ordre de grandeur que le nombre de comparaisons, la complexité de la fonction est donc en $\Theta(n^2)$.

Exercice 2 : Diamètre, rayon et sommets centraux d'un graphe (6 points)

Dans tout cet exercice, on ne considère que des graphes non-orientés connexes.

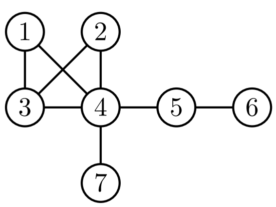
1. Soit G_1 le graphe défini par la matrice sommet-sommet suivante :

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

(a) Dessinez le graphe G_1.

(b) Donnez une représentation de G_1 par liste d'adjacence.

Solution:



$[[3,4],[3,4],[1,2,4],[1,2,3,5,7],[4,6],[5],[4]]$

Soit $G = (V, E)$ un graphe connexe. On rappelle que la distance $d(u, v)$ entre deux sommets u et v est le nombre d'arêtes d'une plus courte chaîne entre u et v.

On définit l'**excentricité** $\varepsilon(u)$ d'un sommet u de G comme la distance de u au sommet $v \in V$ le plus éloigné de u. Plus formellement, $\varepsilon(u) = \max_{v \in V} d(u, v)$.

Le **diamètre** $\delta(G)$ de G est l'excentricité maximum d'un sommet. Plus formellement, $\delta(G) = \max_{u \in V} \varepsilon(u)$. Deux sommets à distance $\delta(G)$ l'un de l'autre sont dits **diamétralement opposés**.

À l'inverse, un **sommet central** est un sommet d'excentricité minimum. L'excentricité d'un sommet central est appelée le **rayon** $r(G)$ du graphe. Ainsi, $r(G) = \min_{u \in V} \varepsilon(u)$.

2. Cette question porte à nouveau sur le graphe G_1 de la question 1.
   (a) Donnez une plus courte chaîne entre les sommets 1 et 5. À quelle distance sont-ils ?
   (b) Quel est le sommet le plus loin de 2 ? Quelle est l'excentricité de 2 ?
   (c) Quel est le diamètre de G_1 ? Donnez deux sommets diamétralement opposés.
   (d) Quel est le rayon de G_1. Donnez un sommet central. Est-il unique ?

Solution:
(a) La plus courte chaîne est $1 \to 4 \to 5$, il a longueur 2 et les sommets 1 et 5 sont donc à distance 2.
(b) Le sommet le plus éloigné de 2 est le sommet 6, qui est à distance 3. L'excentricité de 2 est donc 3.
(c) Le diamètre du graphe est 3. Les sommets 2 et 6 sont diamétralement opposés.
(d) Le rayon du graphe est 2. Il y a deux sommets centraux : 4 et 5.

3. (a) Quel parcours pouvez-vous utilisez pour connaître l'excentricité d'un sommet u ? Justifiez.
   (b) Donnez une méthode simple pour calculer le rayon d'un graphe. Donnez sa complexité en fonction de la complexité $c(n, m)$ du parcours que vous avez choisi à la question précédente.

Solution:
(a) Le parcours en largeur depuis u donne la distance de u à tous les autres sommets du graphe. Le maximum de ces distances (la distance du dernier sommet visité) est l'excentricité de u.
(b) On peut réaliser un parcours depuis chaque sommet du graphe pour connaitre l'excentricité de tous, et sélectionner le minimum. La complexité est en $n \times c(n, m)$.

4. (a) Prouver que le diamètre d'un graphe est toujours inférieur ou égal au double de son rayon.
   (b) Donnez un graphe dont le diamètre est égal au double du rayon.
   (c) Le diamètre d'un graphe peut-il être égal à son rayon ? Justifiez.

Solution:
(a) Soient G un graphe, d son diamètre et r son rayon. Soient u et v deux sommets diamétralement opposés et c un centre du graphe. Soit C_1 un plus court chemin de u à c. Par définition, la longueur de ce chemin est au plus l'excentricité de c, qui est égale à r. Soit C_2 un plus court chemin de c à v. On sait de même que sa longueur est inférieure ou égale à r. En concaténant C_1 et C_2, on obtient un chemin C de u à v de longueur $l \leq 2r$. Comme C est un chemin de u à v, on sait que sa longueur est au moins celle d'un plus court chemin de u à v, soit d. D'où $2r \geq l \geq d$.
(b) On considère le graphe $(\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}\})$. Son rayon est 1 (le sommet central est 2) et le diamètre est 2 (1 et 3 sont diamétralement opposés).
(c) Par exemple, un graphe à 2 sommets avec une arête entre eux. Son diamètre et son rayon sont 1.

Exercice 3 : QCM (5 points) Un seul choix est possible. Une bonne réponse = $\frac{1}{2}$ points. Une mauvaise réponse = -$\frac{1}{4}$ points. Si la note globale du QCM est négative, on la considère égale à 0.

1. Pour démontrer $a \Rightarrow b$ par en utilisant la contraposée, il faut :
   - ◯ démontrer que si a est faux, alors b est faux ;
   - √ **démontrer que si b est faux, alors a est faux ;**
   - ◯ démontrer que l'on si b est faux, alors a est vraie ;
   - ◯ Aucun des choix précédents.
2. Une seule de ces affirmations est vraie, laquelle ?
   - √ **Pour un tableau de n entiers trié en ordre croissant, la complexité du tri par insertion est en $\Theta(n)$;**
   - ◯ Pour un tableau de n entiers trié en ordre croissant, la complexité du tri à bulles est en $\Theta(n)$;
   - ◯ Pour un tableau de n entiers trié en ordre croissant, la complexité du tri par sélection est en $\Theta(n)$;
   - ◯ Les trois affirmations précédentes sont fausses.
3. Soit T un arbre binaire ayant 5 noeuds à deux fils et 3 noeuds à un fils. Une seule de ces affirmations est vraie, laquelle ?
   - ◯ La taille de T est 11 ;
   - ◯ La taille de T est 8 ;
   - √ **La taille de T est 14 ;**
   - ◯ Plusieurs valeurs sont possibles pour la taille de T.
4. L'ABR obtenu par insertions successives des clefs 7, 3, 10, 2, 9 et 5 a pour parcours préfixe
   - ◯ $(7, 2, 5, 3, 9, 10)$;
   - ◯ $(7, 3, 2, 5, 9, 10)$;
   - √ $(7, 3, 2, 5, 10, 9)$ **;**
   - ◯ Aucune des réponses précédentes.
5. On considère l'ABR obtenu par insertions successives des clefs 10, 7, 15, 12, 9, 2, 18, et 11. Quel est le parcours préfixe de l'ABR obtenu quand on a supprimé la clef 15 en utilisant l'algorithme de suppression du cours ?
   - ◯ $(10, 7, 2, 9, 18, 11, 12)$;
   - √ $(10, 7, 2, 9, 12, 11, 18)$ **;**
   - ◯ $(10, 7, 2, 9, 11, 12, 18)$;
   - ◯ Aucune des réponses précédentes.
6. La taille maximale d'un arbre binaire de hauteur h est égale à
   - ◯ $2^{h+1} - 1$;
   - ◯ $2 \times h + 1$;
   - √ $2^h - 1$ **;**
   - ◯ Aucune des réponses précédentes.
7. On souhaite insérer la clefs 1 dans le tas $T = (9, 3, 5, 4, 8, 9, 6, 7, 10, 12)$.

- ◯ Après l'insertion, on obtient le tas $(10, 1, 3, 4, 5, 10, 6, 7, 8, 9, 12)$;
- $\surd$ **Après l'insertion, on obtient le tas** $(10, 1, 3, 4, 8, 5, 6, 7, 10, 12, 9)$ **;**
- ◯ C'est impossible car T n'est pas un tas ;
- ◯ Aucune des réponses précédentes.

8. La suppression du minimum du tas $T = (10, 1, 2, 3, 5, 10, 4, 7, 17, 15, 18)$ donne le tas :
   - ◯ $(9, 2, 3, 5, 10, 15, 4, 7, 17, 18)$;
   - ◯ $(9, 2, 5, 3, 10, 15, 4, 7, 18, 17)$;
   - $\surd$ $(9, 2, 5, 3, 15, 10, 4, 7, 17, 18)$ **;**
   - ◯ Aucune des réponses précédentes.
9. Soit $G = (V, E)$ un graphe de 75 arêtes. Soit n sont nombre de sommets. Une seule affirmation est vérifié, laquelle ?
   - ◯ Si $n \leq 75$, alors G est connexe ;
   - ◯ Si G est connexe, alors $n \leq 75$;
   - ◯ Si $n = 75$, alors G est un arbre ;
   - $\surd$ **Les trois affirmations précédentes sont fausses.**
10. Soit $G = (V, E)$ un graphe non orienté. G est un arbre si et seulement si une seule des affirmations suivantes est vérifiée, laquelle ?
    - ◯ G est acyclique ;
    - ◯ G est connexe et il existe une arête $e \in E$ telle que le graphe $G' = (V, E - \{e\})$ n'est plus connexe ;
    - $\surd$ G **est connexe et pour toute arête** $e \in E$**, le graphe** $G' = (V, E - \{e\})$ **n'est plus connexe.**