Examen 2I003

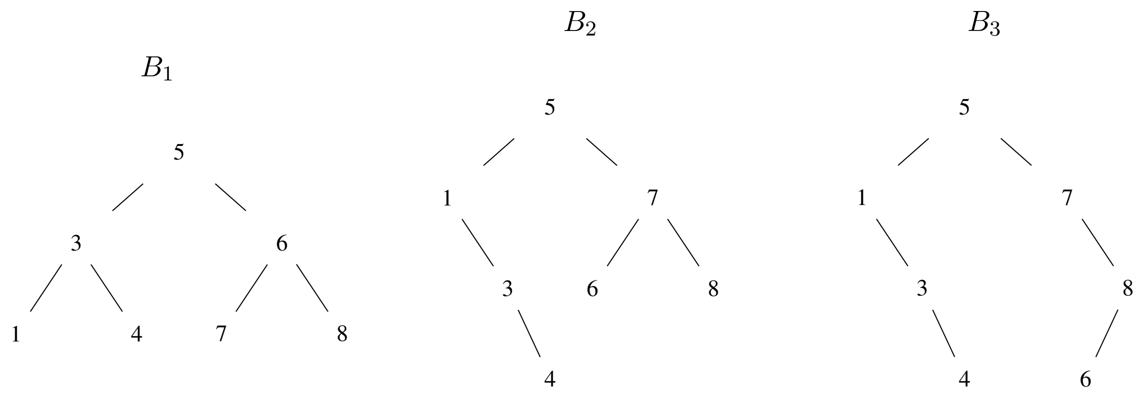
Mercredi 7 Juin 2017, 2 heures
aucun document autorisé

Exercice 1 – Recherche du k-ème plus petit élément - 12 points

Le *k-ème plus petit élément* d'un ensemble E de n nombres tous différents, avec $n \geq k$, est l'élément x de E tel qu'il y a exactement $k-1$ éléments de E strictement inférieurs à x.

Par exemple, le 3-ème plus petit élément de $\{8, 5, 3, 4, 1, 6, 7\}$ est 4.

Recherche dans un ABR



Question 1

1. Rappeler la définition d'un arbre binaire de recherche.
2. Pour chacun des arbres B_1, B_2, B_3, dire s'il est ou non un arbre binaire de recherche. Justifier les réponses négatives.

<u>**Solution**</u>:

1. Voir cours.
2. B_1 n'est pas un ABR car la clef 7 est supérieure à la clef 6.
   B_2 est un ABR.
   B_3 n'est pas un ABR car la clef 6 est inférieure à la clef 7.

Question 2

On considère dans la suite de l'exercice la fonction `ABinfixe(B)` ainsi définie, pour B arbre binaire :

```
def ABinfixe(T):
    if estABvide(T):
        return []
    return ABinfixe(T.gauche) + [T.clef] + ABinfixe(T.droit)
```

Calculer sans justification les valeurs `ABinfixe(B1)`, `ABinfixe(B2)`, `ABinfixe(B3)`.

<u>**Solution**</u>:

```
ABinfixe(B1) : [1, 3, 4, 5, 7, 6, 8]
ABinfixe(B2) : [1, 3, 4, 5, 6, 7, 8]
ABinfixe(B3) : [1, 3, 4, 5, 7, 6, 8]
```

Question 3

Montrer par **induction structurelle** sur B que pour **tout** arbre binaire de **recherche** B, `ABinfixe(B)` est une liste rangée en ordre strictement croissant.

Solution:

Base Si $B = \emptyset$ alors `ABinfixe(B) = []` donc la propriété est vraie.

Induction Soit G et D deux arbres binaires de recherche, supposons que la propriété soit vraie pour G et D. Soit $B = (x, G, D)$ un arbre binaire de recherche alors :
La liste `ABinfixe(B)` est la concaténation de :
- la liste `ABinfixe(G)` qui est rangée en en ordre strictement croissant,
- la liste `[x]`, dont le seul élément x est supérieur à tous les éléments de `ABinfixe(G)`, et inférieur à tous les éléments de `ABinfixe(D)`, puisque B est un ABR,
- la liste `ABinfixe(D)` qui est rangée en en ordre strictement croissant.

Donc la liste `ABinfixe(B)` est rangée en ordre strictement croissant.

Conclusion La propriété est vraie pour tout arbre binaire de recherche.

Question 4

1. En déduire un algorithme qui calcule la k-ème plus petite clef d'un arbre binaire de recherche B.
2. Calculer le nombre de concaténations de listes effectuées par l'appel `ABinfixe(B)`, pour un arbre binaire B de taille n. En supposant que la complexité de la concaténation de deux listes est en $\Theta(1)$, calculer la complexité de `ABinfixe(B)`, pour un arbre binaire B de taille n.
3. En déduire la complexité de la recherche de la k-ème plus petite clef d'un arbre binaire de recherche B de taille n.

Solution:

1. `ABinfixe(B)[k]`
2. Le nombre de concaténations est $2n$. On peut le prouver par induction : $c(B) = 0$ si B est vide, $c(B) = c(G) + c(D) + 2 = 2n(G) + 2n(D) + 2 = 2n(B)$ si $B = (x, G, D)$.

   La complexité de `ABinfixe(B)` est en $\Theta(n)$.
3. Il faut calculer la complexité de `ABinfixe(B)[k]` : la complexité de `ABinfixe(B)` est en $\Theta(n)$ et la complexité de l'accès à $L[k]$ pour une liste de taille n est en $O(n)$, quelle que soit la représentation de la liste. La complexité totale est donc en $\Theta(n)$.

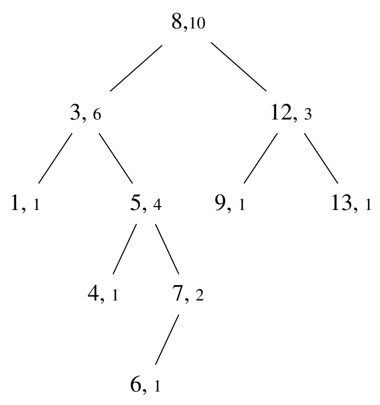
Recherche dans un ABR dénombré

Un arbre binaire de recherche *dénombré* est un arbre binaire de recherche dont chaque nœud x contient deux informations :

— la *clef* (`B.clef`), sur laquelle est effectuée la recherche,

— la *taille* (`B.taille`), qui est égale au nombre de nœuds de l'arbre enraciné en x.

On considère l'arbre binaire de recherche dénombré Bex :



On définit une fonction `taille` sur les arbres binaires dénombrés, dont la complexité est en $\Theta(1)$:

```
def taille(T):
  if estABvide(T):
    return 0
  return T.taille
```

On définit une fonction `ABRD_k_eme(B, k)` sur les arbres binaires de recherche dénombrés :

```
def ABRD_k_eme(B, k):
    """hypothèse␣:␣k␣<=␣taille(T)"""
    print("appel␣avec␣k␣=␣", k)
    t = taille(B.gauche)
    if k == t + 1 :
        res = B.clef
    elif k <= t:
        res = ABRD_k_eme(B.gauche, k)
    else:
        res = ABRD_k_eme(B.droit, k - (t + 1))
    print("retour␣:␣", res)
    return res
```

Question 5

Exécuter l'appel de `ABRD_k_eme(Bex, 6)`, en donnant les affichages successifs et le résultat final.

<u>**Solution**</u>:

```
appel avec k =  6
appel avec k =  6
appel avec k =  4
appel avec k =  2
retour :  7
retour :  7
retour :  7
retour :  7
7
```

Question 6

Montrer par **induction structurelle** sur B que pour **tout** arbre binaire de **recherche dénombré** B et pour tout k tel que $0 < k \leq taille(B)$, `ABRD_k_eme(B, k)` calcule la k-ème plus petite clef de B.

Solution:
Par induction.

Base Si $B = \emptyset$ alors il n'y a aucun k tel que $0 < k \leq taille(B)$ donc la propriété est vraie.

Induction Soit G et D deux arbres binaires de recherche dénombrés, supposons que la propriété soit vraie pour G et pour tout k' tel que $0 < k' \leq taille(G)$ et qu'elle soit vraie pour D et pour tout k' tel que $0 < k' \leq taille(D)$. Soit B un arbre binaire de recherche dénombré de clef x, de sous-arbre gauche G et de sous-arbre droit D. Soit $t = taille(G)$. Trois cas sont possibles.

- $k = t + 1$: dans ce cas, il y a exactement $t = k - 1$ clefs de B qui sont inférieures à x, x est donc la k-ème plus petite clef de B et c'est la valeur calculée par la fonction.
- $k \leq t$: dans ce cas, la k-ème plus petite clef de B est dans le sous-arbre gauche G de B et on peut appliquer l'hypothèse de récurrence, la fonction calcule la k-ème plus petite clef de G, qui est aussi la k-ème plus petite clef de B.
- $k > t + 1$: dans ce cas la k-ème plus petite clef de B est dans le sous-arbre droit D de B et c'est la $k-(t+1)$-ème plus petite clef de D ; par hypothèse de récurrence, l'appel à `ABRD_k_eme(D,k-(t+1))` calcule la $k - (t + 1)$-ème plus petite clef de D, qui est aussi la k-ème plus petite clef de B.

Conclusion La propriété est vraie pour tout arbre binaire de recherche dénombré B et pour tout entier k tel que $0 < k \leq taille(B)$.

Question 7

1. Quelle est la complexité de la recherche de la k-ème plus petite clef dans un arbre de recherche dénombré de taille n dans le meilleur cas ? dans le pire cas ?
2. Mêmes questions pour un arbre de recherche dénombré H-équilibré.

Solution:

1. Dans le meilleur cas, la k-ème plus petite clef est à la racine et la complexité est en $\Omega(1)$. Dans le pire cas l'arbre est filiforme et la complexité est en $O(n)$.
2. Dans le meilleur cas, la k-ème plus petite clef est à la racine et la complexité est en $\Omega(1)$. Dans le pire cas, la k-ème plus petite clef est dans une des feuilles les plus profondes. La hauteur d'un arbre H-équilibré est en $O(\log n)$. La complexité dans le pire cas est donc en $O(\log n)$.

Exercice 2 – Graphes non orientés - 8 points

Dans cet exercice, $G = (V, E)$ est un graphe non orienté. On pose $n = |V|$ et $m = |E|$.

Question 1

Dans cette question, on considère le graphe non orienté $G = (V, E)$ avec $V = \{1, 2, 3, 4, 5, 6\}$ et $E = \{\{1, 2\}, \{2, 3\}, \{3, 1\}, \{4, 5\}, \{5, 6\}\}$. Que valent n et m ? Donnez la matrice sommet-sommet du graphe.

Solution:
$n = 6$, et $m = 5$. La matrice sommet-sommet est :

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

Question 2

Rappelez la définition du degré $d_G(x)$ pour $x \in V$. Quel est le degré des sommets du graphe de la question 1 ?

Solution:
Le degré de tout sommet $x \in V$ est le nombre de sommets de V qui sont adjacents à x. Les degrés du graphe G donné sont :

x	1	2	3	4	5	6
$d_G(x)$	2	2	2	1	2	1

Question 3

Démontrez par récurrence sur le nombre m d'arêtes que, pour tout graphe non orienté $G = (V, E)$, $\sum_{x \in V} d_G(x) = 2m$..

Solution:
La propriété à démontrer s'énonce ainsi, pour $m \geq 0$:

$$\Pi(m) \text{ : pour tout graphe } G = (V, E) \text{ non orienté de } m \text{ arêtes, } \sum_{x \in V} d_G(x) = 2m.$$

La propriété se démontre par récurrence faible.

Base $m = 0$. Pour tout graphe G sans arête, $d_G(x) = 0$ donc $\Pi(0)$ est bien vérifiée.

Induction Soit $m > 0$ tel que $\Pi(m-1)$ soit vérifiée. Soit $G = (V, E)$ un graphe ayant m arêtes, $a = \{u, v\} \in E$, $E' = E \setminus \{a\}$ et $G' = (V, E')$. G' est un graphe qui a $m - 1$ arêtes donc, par hypothèse de récurrence :

$$\sum_{x \in V} d_{G'}(x) = 2(m-1)$$

D'autre part, on observe que $d_G(u) = d_{G'}(u) + 1$, $d_G(v) = d_{G'}(v) + 1$ et que $d_G(x) = d_{G'}(x)$ pour tout $x \in V \setminus \{u, v\}$. On en déduit que

$$\sum_{x \in V} d_G(x) = 2 + \sum_{x \in V} d_{G'}(x) = 2 + 2(m-1) = 2m$$

Donc $\Pi(m)$ est vérifiée.

Conclusion On en conclut que la propriété est vraie pour tout $m \in \mathbb{N}$.

Question 4

1. Qu'est ce qu'un graphe connexe ? Est-ce que le graphe de la question 1 est connexe ?
2. Rappelez la définition de la composante connexe C_x de x pour $x \in V$.
3. Quelles sont les composantes connexes du graphe G de la question 1 ?

Solution:

1. Un graphe est connexe si, il existe une chaîne reliant tout couple de sommets. Le graphe de la question 1 n'est pas connexe.
2. La composante connexe de $x \in V$ est l'ensemble des sommets $y \in V$ tels qu'il existe une chaîne entre x et y.
3. Le graphe est constitué des deux composantes $\{1, 2, 3\}$ et $\{4, 5, 6\}$.

Question 5

Supposons dans cette question que $H = (V, E)$ est un graphe non orienté connexe tel que, $\forall v \in V$, $d(v) \leq 2$. Montrez par récurrence forte sur le nombre de sommets que H est soit un cycle, soit un chaîne.

Solution:
On le montre par récurrence forte sur le nombre de sommets.

Base : Pour $n = 1$, l'unique sommet est de degré 0 et constitue bien une chaîne à lui tout seul.

Induction : Supposons maintenant que la propriété est vérifiée pour tout graphe de strictement moins de n sommets, avec $n > 1$. Soit alors $H = (V, E)$ un graphe de n sommets tel que, $\forall v \in V$, $d(v) \leq 2$. On enlève une arête $e = \{x, y\}$ prise au hasard dans A. On a alors deux cas :

- Si le graphe $H' = (V, E - \setminus\{e\})$ est connexe, il y a donc un chaîne de x à y dans H'. Tous les sommets de cette chaîne, exeptés x et y ont un degré de 2. De plus, dans H, x et y ont un degré de 2. Donc, il n'y a pas d'autres arêtes dans H et H est alors un cycle.
- Sinon, on a alors deux composantes connexes C_x et C_y, chacune d'elles étant un cycle ou une chaîne par récurrence forte. De plus, les degrés de x et y sont au plus de 1, ce sont donc des chaînes. Donc, H est une chaîne.