Préparation à l'oral 1 : arbres et graphes

19 et 21 mai

1 Arbres

$\boxed{1}$

Exercice 1. Dans cet exercice, on autorise les doublons dans un arbre binaire de recherche et pour le cas d'égalité on choisira le sous-arbre gauche. On ne cherchera pas à équilibrer les arbres.

1. Rappeler la définition d'un arbre binaire de recherche.
2. Insérer successivement et une à une dans un arbre binaire de recherche initialement vide toutes les lettres du mot *bacddabdbae*, en utilisant l'ordre alphabétique sur les lettres. Quelle est la hauteur de l'arbre ainsi obtenu ?
3. Montrer que le parcours en profondeur infixe d'un arbre binaire de recherche de lettres est un mot dont les lettres sont rangées dans l'ordre croissant. On pourra procéder par induction structurelle.
4. Proposer un algorithme qui permet de compter le nombre d'occurrences d'une lettre dans un arbre binaire de recherche de lettres. Quelle est sa complexité ?
5. On souhaite supprimer *une* occurrence d'une lettre donnée d'un arbre binaire de recherche de lettres. Expliquer le principe d'un algorithme permettant de résoudre ce problème et le mettre en œuvre sur l'arbre obtenu à la question 2. en supprimant successivement une occurrence des lettres e, b, b, c et d. Quelle en est la complexité ?

$\boxed{2}$ On considère ici des arbres binaires stricts (chaque noeud possède 0 ou 2 fils). Etant donné un arbre a et un sommet s de a, on définit $p(s,a)$ comme la profondeur de s dans a.

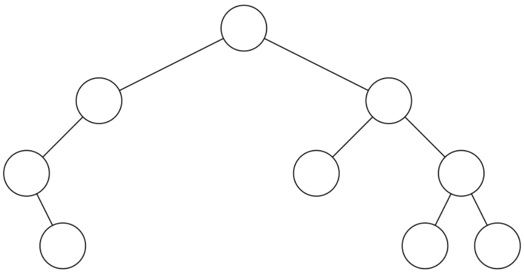
1. Démontrer l'égalité suivante (appelée égalité de Kraft), pour tout arbre binaire strict a : $\sum_{f\in F(a)} 2^{-p(f,a)} = 1$ où $F(a)$ est l'ensemble des feuilles de a.
2. Donner un contre-exemple dans le cas d'un arbre binaire non nécessairement strict.

$\boxed{3}$

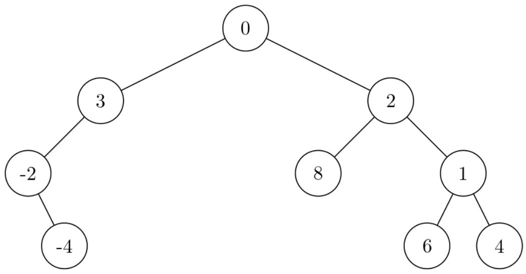
1. On suppose que tous les entiers naturels entre 1 et 1000 sont rangés dans un arbre binaire de recherche. On souhaite retrouver le nombre 444. Parmi les séquences suivantes, lesquelles pourraient être la séquence de noeuds parcourus ? Justifier la réponse.
   — 10, 110, 210, 612, 450, 420, 430, 444 ;
   — 873, 225, 230, 666, 333, 555, 400, 444 ;
   — 448, 441, 447, 442, 446, 443, 445, 444 ;
   — 210, 901, 270, 280, 450, 803, 460, 444 ;
   — 935, 278, 347, 621, 299, 392, 358, 444.
2. Donner un algorithme qui décide si les clés dans un tableau non vide `A[0...(n-1)]` d'entiers peuvent être les clés de noeuds parcourus lors d'une recherche fructueuse dans un arbre binaire de recherche. On suppose que tous les entiers sont compris entre 1 et 1000. L'algorithme doit s'exécuter en $O(n)$.
3. Implémenter cet algorithme en OCAML, à l'aide d'une fonction `parcours_possible : int list -> bool`
4. Ecrire une fonction qui trie un tel tableau (correspondant à une recherche valide dans un ABR).

4

Dans cet exercice, on considère des arbres binaires étiquetés par des entiers relatifs deux à deux distincts. Un nœud est un minimum local d'un arbre si son étiquette est plus petite que celle de son éventuel père et celles de ses éventuels fils. Considérons par exemple l'étiquetage (b) de l'arbre binaire non étiqueté (a) :



(a)



(b)

1. Déterminer le ou les minima locaux de l'arbre (b).
2. Donner une définition inductive permettant de définir les arbres binaires ainsi que la définition de la hauteur d'un arbre. Quelle est la hauteur de l'arbre (b) ?
3. Montrer que tout arbre non vide possède un minimum local.
4. Proposer un algorithme permettant de trouver un minimum local d'un arbre non vide et déterminer sa complexité.

On considère un arbre binaire non étiqueté que l'on souhaite étiqueter par des entiers relatifs distincts deux à deux de manière à maximiser le nombre de minima locaux de cet arbre.

5. Proposer sans justifier un étiquetage de l'arbre (a) qui maximise le nombre de minima locaux.
6. Proposer un algorithme qui, étant donné un arbre binaire non étiqueté en entrée, permet de calculer le nombre maximal de minima locaux qu'il est possible d'obtenir pour cet arbre. Déterminer la complexité de votre algorithme.
7. Montrer que, pour un arbre de taille $n \in \mathbb{N}$, le nombre maximal de minima locaux est majoré par $\left\lfloor \dfrac{2n+1}{3} \right\rfloor$. *On pourra remarquer que les nœuds non minimaux couvrent l'ensemble des arêtes de l'arbre.*

2 Graphes : échauffement

5 Compléter et démontrer la formule suivante : dans un graphe $G = (S, A)$ non orienté, on a $\sum_{s \in S} deg(s) = ...$
Soit $G = (S, A)$ un arbre à n sommets. Combien a-t-il d'arêtes ?
Montrer qu'un arbre avec un sommet de degré 2017 possède au moins 2017 feuilles (une feuille est un sommet de degré 1).

6 Quel est le nombre minimal d'arêtes que peut avoir un graphe connexe à n sommets.
Soit $k \in \mathbb{N}$. Quel est le nombre minimum de composantes connexes d'un graphe à n sommets et $n - k$ arêtes ?

7 Montrer que dans tout graphe avec au moins 2 sommets, il existe 2 sommets de même degré.

8

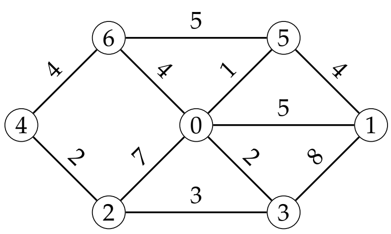
Soit G un graphe dont tous les sommets sont de degré 3. Montrer qu'il a nécessairement un nombre de sommets pair.
Soit n un entier naturel supérieur ou égal à 2. Construire un graphe à $2n$ sommets dont tous les sommets sont de degré 3.

9 Rappeler la définition de la matrice d'adjacence d'un graphe. Soit M cette matrice, que représente M^i (que contiennent ses cases ?) Comment peut-on utiliser cette information pour déterminer si un graphe contient un triangle (un cycle de longueur 3).
Soit G un graphe dont tous les sommets sont de degré de k. Soit M sa matrice d'adjacence dans $(M)_n(\mathbb{R})$. Trouver un vecteur X colonne de taille n tel que $MX = X$.

10 Soit G un graphe non-orienté d'ordre $2p$. On suppose que le degré de chaque sommet est au moins égal à p. Démontrer que ce graphe est connexe.

11

Appliquer l'algorithme de KRUSKAL au graphe suivant et déterminer l'arbre couvrant de poids minimal renvoyé :



3 Graphes

12 Soit $G = (S, A)$ un graphe sans triangle à n sommets.

1. Montrer que pour tout $(x, y \in A), deg(x) + deg(y) \leq n$.
2. Montrer que $\sum_{x \in S} deg(x)^2 = \sum_{(x,y) \in A} deg(x) + deg(y)$.
3. En déduire que $\sum_{x \in S} deg(x)^2 \leq |A|.n$.
4. En déduire avec l'inégalité de Cauchy Shwarz que $(\sum_{x \in S} deg(x))^2 \leq |A|.n^2$.
5. En déduire que $|A| \leq \frac{n^2}{4}$.

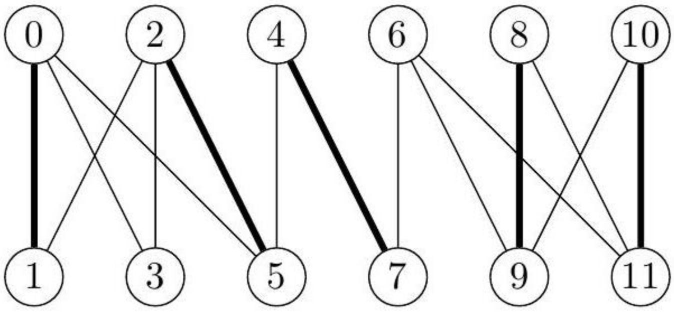
13 Dans cet exercice, on considère uniquement des graphes non orientés.

1. Rappeler la définition d'un couplage et d'un chemin augmentant.
2. Un couplage maximal pour l'inclusion peut-il ne pas être de cardinal maximum ? Un couplage de cardinal maximum peut-il ne pas être maximal pour l'inclusion ? Justifier.

Soit $G = (S, A)$ un graphe biparti d'ordre $n = |S|$ et C un couplage. On suppose que $S = \{0, \ldots, n-1\}$ et que S se décompose en deux ensembles $S = X \sqcup Y$ tels qu'il n'existe aucune arête entre deux sommets de X ou deux sommets de Y. On suppose de plus qu'il existe dans G un chemin augmentant relativement à C. On considère l'algorithme suivant :

```
1  Début algorithme
2      F ← file vide.
3      P ← tableau contenant n fois la valeur −1.
4      Pour x ∈ X Faire
5          Si x est non couvert par C (c'est-à-dire n'appartient à aucune arête de C) Alors
6              Enfiler x dans F.
7              P[x] ← −2.
8      Tant que F est non vide. Faire
9          s ← élément défilé de F.
10         Pour t voisin de s Faire
11             Si P[t] = −1 Alors
12                 Si (s ∈ X et {s, t} ∉ C) ou (s ∈ Y et {s, t} ∈ C) Alors
13                     Enfiler t dans F.
14                     P[t] ← s.
15     Renvoyer P.
```

On considère de plus le graphe biparti G_0 suivant et le couplage C_0 dont les arêtes sont représentées en gras : $C_0 = \{\{0,1\},\{2,5\},\{4,7\},\{8,9\},\{10,11\}\}$:



3. Appliquer l'algorithme au graphe G_0 et donner sans justification le tableau P ainsi obtenu. On supposera que X correspond aux sommets pairs et Y aux sommets impairs. Si un sommet possède plusieurs voisins, on les parcourra par ordre croissant de numéro.
4. Expliquer comment trouver un chemin augmentant en utilisant le tableau P. Déterminer un chemin augmentant dans le graphe G_0 précédent.
5. Avec quelle complexité peut-on réaliser les tests lignes 5 et 12 ? On détaillera la manière de représenter le graphe et le couplage pour réaliser ces opérations.
6. Donner un algorithme permettant de trouver un couplage de cardinal maximum dans un graphe biparti G en utilisant l'algorithme décrit précédemment. Déterminer avec quelle complexité on peut trouver un couplage de cardinal maximum dans un graphe $G = (S, A)$ en fonction de $|S|$ et $|A|$

14 Montrer que dans tout graphe à 6 sommets, on peut trouver 3 sommets tous adjacents ou 3 sommets sans adjacence.

15

Si G est un graphe non orienté, on appelle chaîne hamiltonienne dans G tout chemin dans G passant une et une seule fois par chaque sommet et cycle hamiltonien toute chaîne hamiltonienne dont l'origine et l'arrivée sont reliées dans G. Un graphe possédant un cycle hamiltonien est dit hamiltonien.

1. Le graphe complet à n sommets est-il hamiltonien ?

On considère un graphe simple non orienté $G = (S, A)$ ayant $n \geq 3$ sommets et tel que pour toute paire de sommets $\{u, v\}$ non adjacents dans G, $\deg(u) + \deg(v) \geq n$. Supposons que G est non hamiltonien.

2. Montrer qu'il existe un graphe $H = (S', A')$ tel que $S = S'$, $A \subset A'$, H est non hamiltonien mais rajouter une arête à H le rend hamiltonien.
3. Montrer que ce graphe H possède une chaîne hamiltonienne, qu'on note $(s_1, ..., s_n)$.
4. Montrer qu'il existe $1 < i < n$ tel que $(s_1, s_i) \in A$ et $(s_{i-1}, s_n) \in A$. On pourra considérer les ensembles $I = \{i \in [\![2, n]\!] \mid (s_1, s_i) \in A\}$ et $J = \{i \in [\![2, n]\!] \mid (s_{i-1}, s_n) \in A\}$.
5. Que déduit-on des questions précédentes quant au caractère hamiltonien ou non de G ? *Indication : faire un dessin !*