POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG2810
STRUCTURES DISCRÈTES

TD 11 : ARBRE

Directives pour la remise:

- La remise est individuelle et se fait à la fin de la séance de TD.
- Répondez directement sur ce document Word (docx).
- Lorsque vous avez terminé, générez un PDF avec le nom sous le format: ***SectionDeTD-Matricule.pdf*** (exemple: *04-1234567.pdf*).
- Téléchargez votre fichier PDF dans la boîte de remise située dans la Zone TDs de la page Moodle du cours.
- Choisissez la boîte de remise qui correspond à votre section de TD.
- Aucun retard et aucune remise par courriel ne seront accepté.
- Dans l'intérêt de l'équité pour tous les étudiants, vous devez modifier le fichier Word (docx) fourni. Modifier le fichier EXCLUT le fait d'intégrer des scans de rédaction manuscrite ou d'y écrire avec un stylet.
- Le non-respect des consignes entraînera automatiquement la note 0 pour ce TD.

Objectifs du TD11

Exercice 1: Dénombrer les éléments d’un arbre m-aire complet.

Exercice 2: Modéliser un problème dans lequel une série de décisions conduit à une solution à l'aide d'arbres.

Exercice 3: S’exercer à évaluer des expressions en notation polonaise ou polonaise inverse.

Exercice 4: Résoudre un problème en utilisant un parcours d’arbre avec retour en arrière.

Exercice 5: Appliquer l’algorithme de Kruskal afin d’obtenir l’arbre couvrant de poids minimum d’un graphe pondéré.

Exercice 6: Démontrer une proposition importante de la preuve que l’algorithme de Kruskal est correct.

Exercice 7: Appliquer les notions d’arbre m-aire complet aux arbres d’opérations.

Exercice 8: Proposer un algorithme qui permet de parcourir dans l’ordre un arbre binaire de recherche.

La remise est individuelle mais le travail en équipe est encouragé. Veuillez inscrire votre nom, prénom et matricule ainsi que les noms des collègues avec lesquels vous avez collaboré pour le TD.

Nom:

Prénom:

Matricule:

Collègues:

Exercice 1

Combien de sommets et combien de feuilles possède au maximum un arbre m-aire complet de hauteur h ?

Réponse:

Cet arbre a 1 sommet au niveau 0, m sommets au niveau 1, m^2 sommets au niveau 2, . . . , m^h sommets au niveau h. Il a donc:

$$1 + m + m^2 + \cdots + m^h = \frac{m^{h+1} - 1}{m - 1}$$

sommets en tout. Les sommets au niveau h sont les seules feuilles, et donc il y a m^h feuilles.

Exercice 2

Il y a n pièces de monnaie numéroté 1, 2, …, n parmi lesquelles une est fausse. La fausse pièce pèse légèrement moins que les pièces authentiques.

a) Si $n = 3$, quel est le nombre minimum de pesées nécessaires pour identifier (dans tous les cas) la fausse pièce. Détaillez votre raisonnement.

Réponse:

Une seule pesée. Nous pouvons mettre la pièce 1 du côté droit et la pièce 2 du côté gauche. Si la pièce 1 est plus légère alors nous avons identifié la fausse pièce, si la pièce 2 est plus légère alors nous avons identifié la fausse pièce et si les deux pièces sont de même poids alors la troisième pièce est la fausse pièce.

b) Si $n = 4$, quel est le nombre minimum de pesées nécessaires pour identifier (dans tous les cas) la fausse pièce. Détaillez votre raisonnement.

Deux pesées sont nécessaires. Nous pouvons mettre la pièce 1 et 2 du côté droit et la pièce 3 et 4 du côté gauche. Si la pièce 1 et 2 sont plus légères alors nous pouvons faire une seconde pesée avec la pièce 1 du côté gauche et la pièce 2 du côté droit afin de déterminer la fausse pièce qui sera plus légère. La même procédure peut être utilisé si ce sont les pièces 3 et 4 qui sont plus légères lors de la première pesée.

c) En vous servant des propriétés des arbres m-aires complets, donnez une borne inférieure au nombre de pesées nécessaires pour identifier (dans tous les cas) la fausse pièce parmi n pièces de monnaie.

Réponse:

Puisqu'une pièce est fausse parmi n, il existe alors n possibilités.

Suite à une pesée nous pouvons déterminer si la fausse pièce est à gauche, à droite ou n'est pas parmi les pièces sur la balance. Par conséquent, l'arbre de décision pour la séquence de pesées est un arbre 3-aire.

Il y a au moins n feuilles dans l'arbre de décision car il y a n résultats possibles et chaque résultat possible doit être représenté par au moins une feuille.

Dans le pire cas, le nombre de pesées nécessaires pour déterminer la pièce contrefaite est la hauteur de l'arbre de décision. L'arbre de décision ne peut pas être de hauteur plus petite que $\lceil \log_3 n \rceil$.

Nous pouvons donc donner la borne inférieure $\lceil \log_3 n \rceil$

Exercice 3

Quelle sont les valeurs de ces expressions postfixées (polonaise inverse) ? Détaillez vos calculs.

a) $5\ 2\ 1-\ -\ 3\ 1\ 4\ +\ +\ *$

Réponse:

Nous détaillons les réponses en montrant entre parenthèses l'opération qui est appliquée, en travaillant de gauche à
à droite (Ceci implique toujours la première occurrence d'un symbole d'opérateur).

$5(21-)-314++* = (51-)314++* = 43(14+)+* = 4(35+)* = (48*) = 32$

b) $9\ 3\ /\ 5\ +\ 7\ 2\ -\ *$

Réponse:

$(93/)5+72-* = (35+)72-* = 8(72-)* = (85*) = 40$

Exercice 4

Il existe des problèmes qui ne peuvent être résolus qu'en effectuant une recherche exhaustive de toutes les solutions possibles. Une façon de rechercher systématiquement une solution est d'utiliser un arbre de décision, où chaque sommet interne représente une décision et chaque feuille une solution possible.

Ainsi le retour en arrière (*backtracking* en anglais) est une famille d'algorithme qui permet de tester systématiquement l'ensemble des possibilités. Vous pouvez alors faire une recherche en profondeur jusqu'à atteindre un cul-de-sac et alors faire un retour sur trace afin de visiter un nouvel embranchement. La procédure se poursuit jusqu'à ce qu'une solution soit trouvée ou qu'il soit établi qu'aucune solution n'existe.

Utilisez le retour en arrière pour trouver un sous-ensemble, s'il existe, de l'ensemble $\{27, 24, 19, 14, 11, 8\}$ avec la somme :

a) 20

Réponse:

Le plus grand nombre qui peut éventuellement être inclus est 19. Étant donné que la somme de 19 et tout nombre plus petit dans la liste est supérieur à 20, nous concluons qu'aucun sous-ensemble avec la somme 20 ne contient 19. Ensuite, nous essayons 14 et arrivons à la même conclusion . Enfin, nous essayons 11, et notons qu'après avoir inclus 8, la liste est épuisée et la somme n'est pas 20. Il n'y a donc pas de sous-ensemble dont la somme est 20.

b) 41

Réponse:

En commençant par 27 dans l'ensemble, nous constatons rapidement que le sous-ensemble {27,14} a la somme souhaitée de 41.

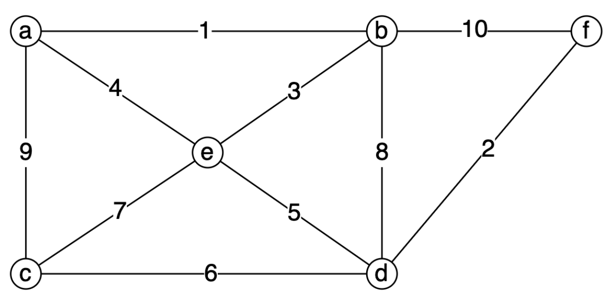
c) 60

Réponse:

Nous essayons d'abord de mettre 27 dans le sous-ensemble. Si nous incluons également 24, aucun autre ajout n'est possible, donc nous revenons en arrière et essayons d'inclure 19 avec 27. Il est maintenant possible d'ajouter 14, ce qui nous donne la somme souhaitée de 60.

Exercice 5

Utilisez l'algorithme de Kruskal pour trouver un arbre couvrant de poids minimum pour le graphe pondéré suivant.



Réponse:

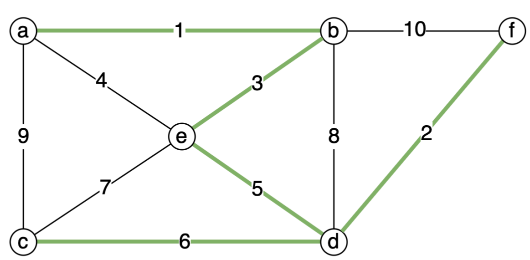
1. Les arêtes sont d'abord triées en ordre croissant de poids:

{a, b} : 1
{d, f} : 2
{b, e} : 3
{a, e}: 4
{d, e} : 5
{d, c} : 6
{c, e} : 7
{b, d} : 8
{a, c} : 9

2. Les arêtes qui ne forment pas de cycle sont ajoutés successivement dans l'ordre.

3. Le graphe possède 6 sommets et l'arbre couvrant de poids minimum possède les 5 arêtes :

{a, b}, {d, f}, {b, e}, {d, e}, {d, c}.



Exercice 6

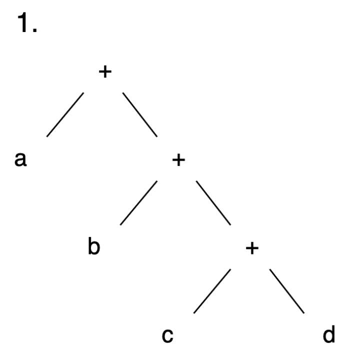
Démontrez que si G est un graphe pondéré avec des poids d'arêtes distincts, alors pour chaque circuit de G, l'arête de poids maximum dans ce circuit n'appartient à aucun arbre couvrant de poids minimum de G.

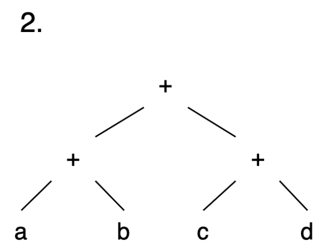
Réponse:

Supposons qu'un arbre couvrant minimum T contienne l'arête e = {u, v} qui est l'arête de poids maximum dans le circuit simple C. Supprimez e de T. Cela crée une forêt avec deux composants, l'un contenant u et l'autre contenant v. Suivez les arêtes du chemin C, partant de u. À un moment donné, ce chemin doit sauter de la composante de T-e contenant u à la composante de T-e contenant v, disons en utilisant l'arête f. Cette arête ne peut pas être dans T, car e peut être la seule arête de T joignant deux composantes (sinon il y aurait un circuit dans T). Parce que e est l'arête de plus grand poids, dans C, le poids de f est plus petit. L'arbre formé en remplaçant e par f dans T a donc un poids plus faible, une contradiction.

Exercice 7

L'opération d'addition est associative. Pour cette raison, plusieurs arbres d'opérations peuvent représenter l'expression $a + b + c + d$.





Supposons qu'un processeur puisse faire une opération par cycle d'horloge. Si vous pouvez effectuer des opérations en parallèle, combien de cycle d'horloge avez vous besoin pour obtenir le résultat:

a) En suivant la démarche indiqué par l'arbre d'opérations 1.
Réponse:

3 cycles

b) En suivant la démarche indiqué par l'arbre d'opérations 2.

Réponse:

2 cycles

c) Combien de cycles d'horloge avez vous besoin au minimum pour effectuer 2^n opérations d'additions, en supposant que vous disposiez d'une capacité de calcul parallèle adéquate.

Réponse:

n cycles

Exercice 8

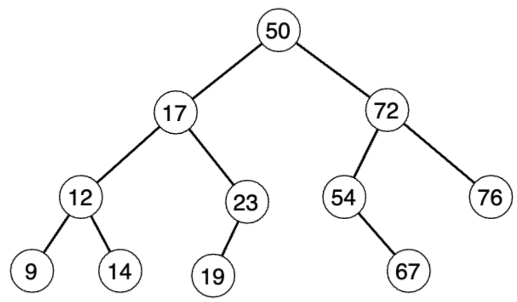
a) Proposez un algorithme afin d'imprimer dans l'ordre infixé la valeur associée aux sommets d'un arbre binaire.

Réponse:

Voici une version récursive de l'algorithme du parcours d'un arbre dans l'ordre infixé:

```
parcours_infixé(x) {
	if (x != null) {
		parcours_infixé(gauche(x));
		imprimer(x);
		parcours_infixé(droit(x));
	}
}
```

b) Remarquez pour l'arbre binaire de recherche suivant, que tout sommet a un enfant de gauche de valeur plus petite que lui-même et un enfant de droite de valeur plus grande que lui-même. Quel sera le résultat de votre algorithme lors du parcours de cet arbre binaire?



Réponse:

Les sommets seront visités dans l'ordre infixé et les valeurs associées aux sommets seront imprimées dans l'ordre suivant:

9, 12, 14, 17, 19, 23, 50, 54, 67, 72, 76.