Révision 3

1 Programmation dynamique

1.1 Exercise 1

En utilisant l'algorithme de programmtion dynamique d'un arbre statique pour minimer le temps de recherche, donnez les matrices C et **racine** et donner un arbre optimal pour les clés suivantes avec leur temps de recherche :

k_i	1	2	3	4	5
P_i	0.10	$0,15$	$0,25$	$0,20$	$0,30$

1.2 Exercise 2

Refaites la même chose qu'à l'exercise 1, mais avec les données suivantes :

k_i	1	2	3	4	5	6
P_i	0.22	$0,10$	$0,20$	$0,10$	$0,15$	$0,23$

2 Algorithme glouton

2.1 Exercise 1

Trouver un encodage optimal pour les lettres suivantes avec leurs fréquences en utilisant le code Huffmann.

k_i	a	b	c	d	f	e
P_i	0.08	$0,15$	$0,17$	$0,20$	$0,15$	$0,25$

2.2 Exercise 2

Déterminer la valeur et le poids du sac alpin de taille 526 si on a les objets suivants avec leur poids et leur valeurs :

i	1	2	3	4	5	6
p_i	25	20	20	40	20	40
v_i	75	45	24	90	30	88

2.3 Exercise 3

Déterminer les pièces à échanger pour le montant en utilisant l'algorithme glouton de la petite monnaie pour les instances suivantes :

1. $p = [1, 3, 5, 7, 10, 15]$ et $m = 102$
2. $p = [2, 4, 5, 6, 10, 25]$ et $m = 215$

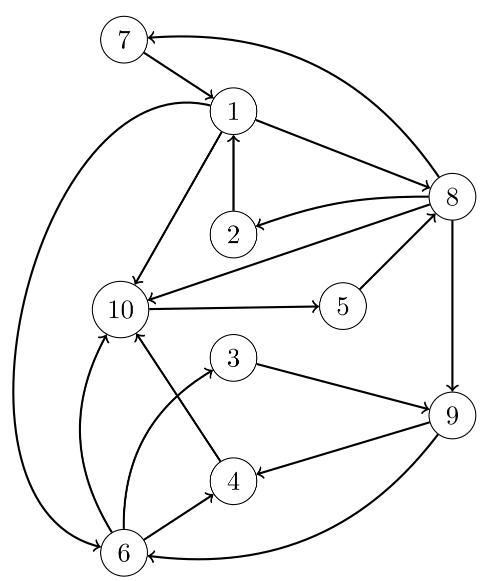
Dans le cas où il n'y a pas de solution, mais qu'il y a une solution possible ou une meilleure solution optimale, alors cités la solution optimale.

3 Algorithmes sur les graphes et retour arrière

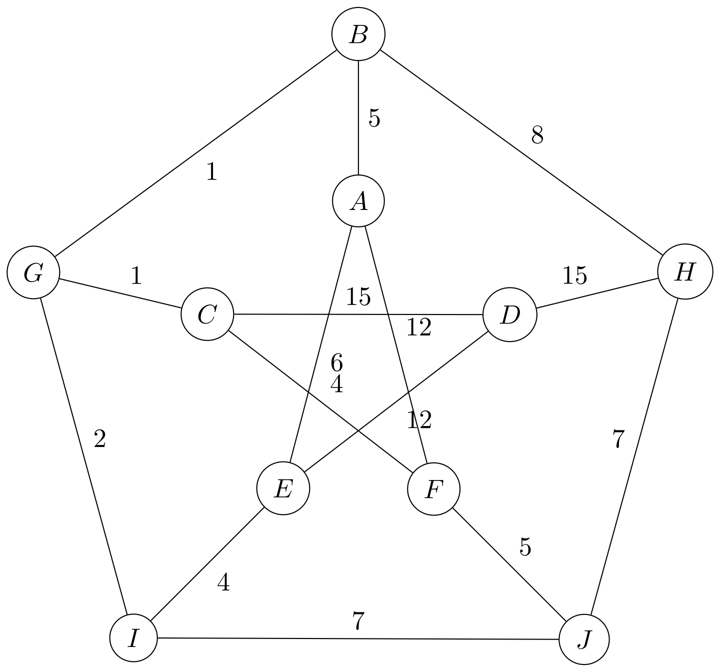
3.1 Exercise 1

Faite un parcours en largeur et en profondeur pour les graphes suivants :

Point de départ : 1.



Point de départ : A

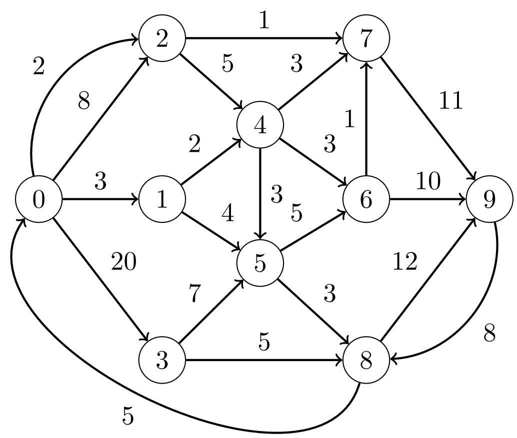


3.2 Exercise 2

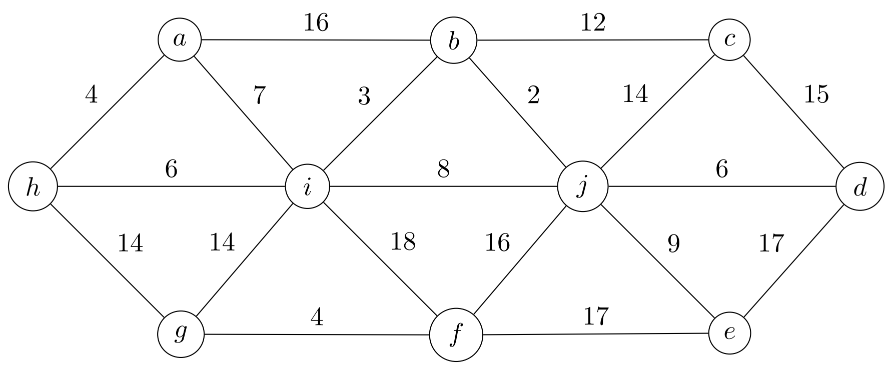
3.2.1 Question A

Trouver un arbre de recouvrement minimale pour les deux graphes suivantes en utilisant l'algorithme de Prim et de Kruskal.

Pour ce graphe, commencez par le sommet 0 pour l'algorithme de Prim.



Pour ce graphe, commencez par le sommet f pour l'algorithme de Prim.



3.3 Question B

Si on fait un des deux algorithmes pour trouver un arbre de recouvrement minimale sur un graphe qui est un arbre, quel sera le résultat ?

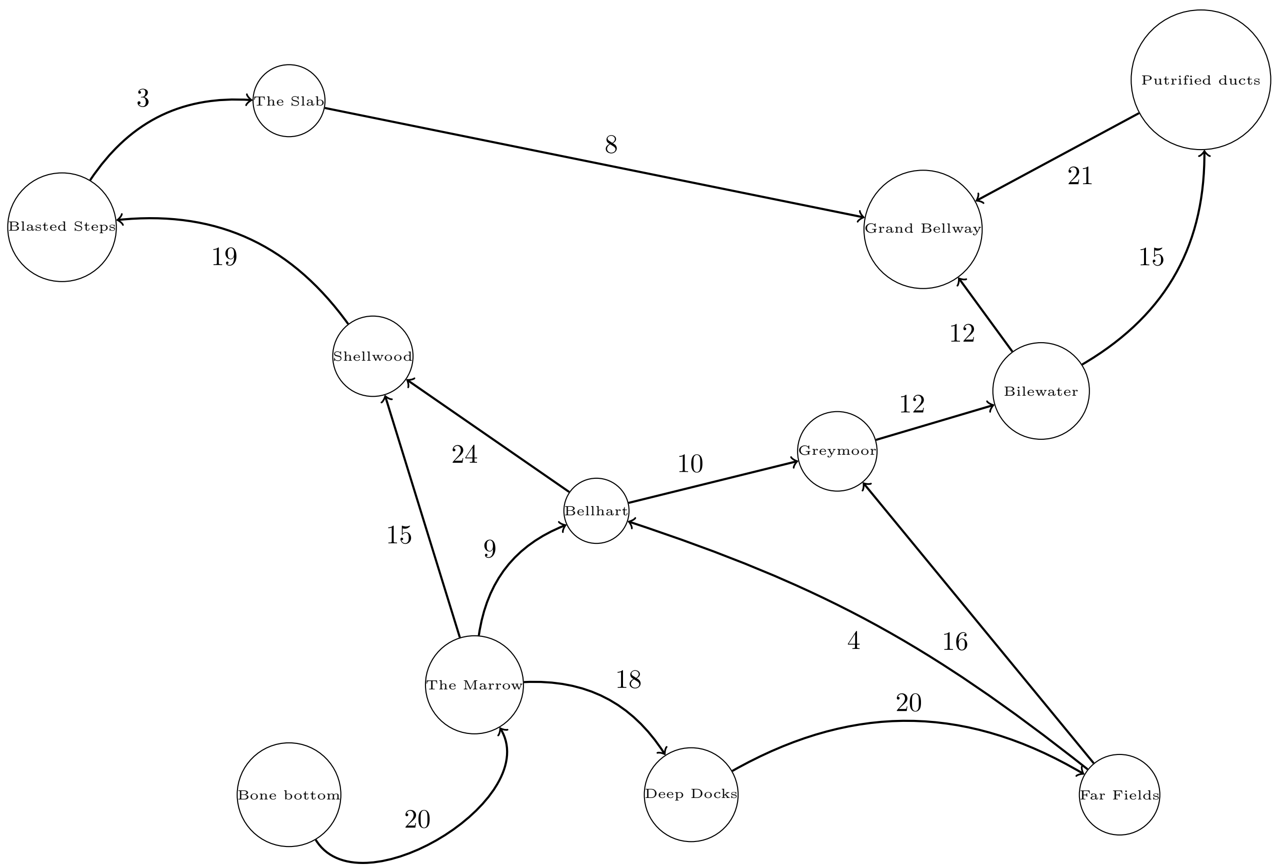
Rappel : un graphe est un arbre s'il répond aux critères suivants :

1. Le graphe est acyclique
2. $|S| = |A| + 1$

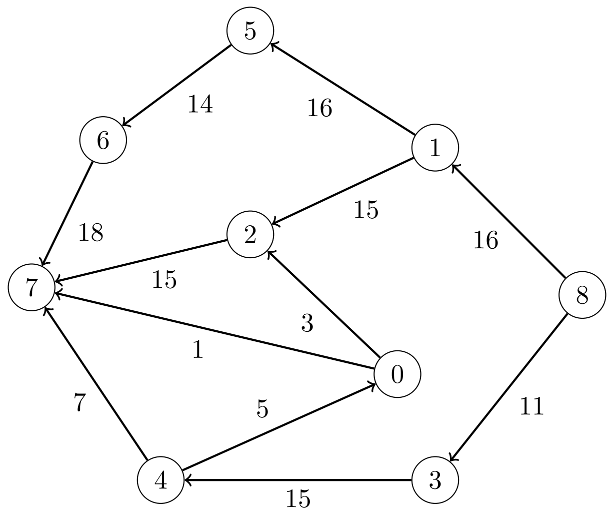
3.4 Exercise 3

Trouver le flot maximal en utilisant l'algorithme de Ford-Fulkerson avec la variante Edmond-Karp pour les graphes suivants :

Sommet source : **Bone bottom**
Sommet puit : **Grand Bellway**



Sommet source : 8
Sommet puit : 6



3.5 Exercise 4

On doit répartir 4 tâches en affectant chacune de ces tâches à un agent différent choisi parmi un ensemble de 4 agents. Pour chaque agent, on connaît le coût de chacune des tâches :

Agents/Tâches	1	2	3	4
1	20	25	10	25
2	35	18	20	15
3	15	10	20	30
4	10	14	25	10

a) Trouvez la solution optimale en appliquant une stratégie de séparation et d'évaluation, en initialisant le score avec une affectation diagonale.

b) Combien de feuilles auriez-vous explorées en initialisant le score avec la stratégie gloutonne suggérée à la page 59 du chapitre sur les algorithmes sur les graphes ?

3.6 Bonus

3.6.1 Question 1

Refaites l'exercise 3, mais avec une recherche en pronfondeur.

4 Np-complétude

4.1 Exercise 1

Convertissz l'intances suivante du problème SAT en une instance du problème 3-FNC-SAT :

$$\phi = (x_3 \oplus \neg x_1) \wedge \neg(x_1 \rightarrow (\neg x_2 \wedge x_3)) \vee x_2$$

4.2 Exercise 2

Soit les instances d'un problème 2-FNC-SAT suivantes, vérifier que ces instances sont satisfaisables. S'il ne sont pas satisfaisable, citez une composante fortement connexe qui contient une variable et son inverse dans le même composante.

a) $(x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_4 \vee \neg x_3) \wedge (x_1 \vee x_4) \wedge (\neg x_1 \vee \neg x_4)$

b) $(x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (x_5 \vee x_6) \wedge (\neg x_5 \vee x_1)$

4.3 Exercise 3

Soit l'instance 3-FNC-SAT suivante :

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$$

convertissez cette instance en une instance Clique et trouvez une Clique.

4.4 Bonus

4.4.1 Réduction SAT vers Clique

Soit une instance du problème SAT suivante :

$$\phi = (x_1 \odot x_2) \wedge (x_2 \rightarrow x_3)$$

trouver une Clique valide en montrant l'ensemble des clauses après la réduction en 3-FNC-SAT pour ensuite la réduire en Clique.

Note :

a	b	$a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

5 Analyse amortie (Complémentaire)

5.1 Question 1

Expliquez en vos mots

1. Ce qu'est l'analyse amortie
2. Les différentes méthodes utilsés pour faire cette analyse.

5.2 Question 2

Expliquer pourquoi l'analyse amortie peut donner un aperçu plus réaliste de la performance d'une structure de données en effectuant cette analyse que plutôt une analyse classique.

5.3 Question 3

Outre la mémoire utilisée, expliquer pourquoi le monceau de Fibonnacci peut être plus lent que d'autre monceau malgré que ses opérations sont presque tous exécutée en temps constant amortie.

5.4 Question 4

Lors du redimenssionnement d'un tableau dynamique, si la nouvelle dimension du tableau augmentait de 1 au lieu de doubler comparé à la taille de l'ancien tableau, est-ce que faire l'analyse amortie aurait donnée le même résultat ?

5.5 Question 5

Quel serait le nombre minimum de clés qu'un arbre de degré 4 devrait contenir dans un monceau de Fibonnacci ?

5.6 Hachage dynamique

Dans certains langages de programmation, les modules, bibliothèques implémentant des tables des hachages sont des tables de hachages dont leur taille double (ou au moins géométriquement) sous certaines conditions. Cette condition est que si α une constante de rapport entre le nombre de clés et de la taille de la table de hachage atteint un certain seuil, alors lors de la prochaine insertion d'une nouvelle clé fera en sorte que le nombre de casiers sera doublé et les clés déjà présents seront rehachés et déplacer dans leurs cases respectifs avant d'insérer la nouvelle clé.

Le cas est aussi applicable dans le cas où $\frac{a_{max}}{4} = \alpha$. Dans ce cas-ci, on réduit la taille du tableau par deux et on rehache tous les éléments. En bref,

$$\frac{a_{max}}{4} \leq \alpha = \frac{n}{m} < \alpha_{max}$$

Dans le cas de collisions, il y a deux méthodes possibles (outre le fait de changer la fonction de hachage).

1. Insertion par chaînage : chaque case est une liste chainée de clé est un nouvel élément est mis en tête de la liste.
2. Adressage ouverte : Lors de l'insertion d'un élément, on insére l'élément à une case voisine qui ne contient pas d'élément.

En prenant une des méthodes de gestion de collisions et en supposant que la fonction de hachage permet de disperser le plus possible les clés, faites l'analyse amortie pour les opérations suivants :

1. Inserer une nouvelle clé
2. Regarder si une clé est présente
3. Supprimer une clé

et avec le seuil pour α pour les gestions de collisions suivantes :

1. Insertion par chaînage : 2
2. Adresse ouverte : 0,70

5.7 Autre cas

Pour d'autres études de cas, consulter la série sur l'analyse amortie qui se trouve au labo #10.

6 Théorie de l'information

6.1 Exercise 1

Soit les lettres suivantes avec leur fréquences :

Lettres	a	b	c	d	e	f	g
Fréquences	$0,05$	$0,25$	$0,13$	$0,12$	$0,10$	$0,16$	$0,19$

Trouver un code de longueur minimal ainsi que leur efficacité en utilisant les méthodes suivantes :

1. Longueur fixe
2. Huffmann
3. Shanon-Fanon

6.2 Exercise 2

Soit $x = (x_1, x_2, x_3, x_4)$ un mot auquel on ajoute 3 symboles y_1 et y_2 pour former un mot-code.

La fonction f suivante permet de calculer les symboles de contrôle (addition modulo 2) :

$$\begin{aligned} y_1 &= x_1 \ + \quad +x_3 + x_4 \\ y_2 &= x_1 \ + x_2 \ + x_3 + \end{aligned}$$

D'où :

- l'information $x = (x_1, x_2, x_3, x_4)$.
- la redondance $y = f(x) = (y_1, y_2)$.
- le mot-code $c = (x_1, x_2, x_3, x_4, y_1, y_2)$

1. Quel est l'ensemble des mots-codes ?
2. Déterminer sa longueur, la dimension et la distance du code ?
3. Calculer le taux, la capacité de détection et la capacité de correction du code
4. On reçoit le mot-code 100110 . Une erreur de transmission s'est elle produite ? Si oui, donner son syndrome et dite si on peut le corriger ?

6.3 Exercise 3

Soit $x = (x_1, x_2, x_3, x_4)$ un mot auquel on ajoute 3 symboles y_1, y_2 et y_3 pour former un mot-code.

La fonction f suivante permet de calculer les symboles de contrôle (addition modulo 2) :

$$\begin{aligned} y_1 &= x_1 \ + \quad +x_3 + x_4 \\ y_2 &= x_1 \ + x_2 \ + x_3 + \\ y_3 &= \quad + \quad + x_3 + x_4 \end{aligned}$$

D'où :

- l'information $x = (x_1, x_2, x_3, x_4)$.
- la redondance $y = f(x) = (y_1, y_2, y_3)$.
- le mot-code $c = (x_1, x_2, x_3, x_4, y_1, y_2, y_3)$

1. Quel est l'ensemble des mots-codes ?
2. Déterminer sa longueur, la dimension et la distance du code ?

3. Calculer le taux, la capacité de détection et la capacité de correction du code
4. On reçoit le mot-code 1001101 . Une erreur de transmission s'est elle produite ? Si oui, donner son syndrome et dite si on peut le corriger ?