Complexité d'algorithmes

Question 1. Les tableaux et les listes

Question 1.1 Evaluer la complexité d'un algorithme qui à partir de deux listes triées A et B construit une liste unique contenant les éléments des deux listes A et B.

Question 1.2 Etant donné un tableau trié d'entiers, évaluer la complexité d'un algorithme qui teste si l'entier x est contenu dans le tableau.

Question 1.3 Soit T un tableau de p entiers. Décrire un algorithme naïf qui retourne le plus petit élément, ainsi que le second plus petit élément d'un tableau en $\frac{3p}{2} + O(1)$ comparaisons. ***Indication : diviser le tableau en deux.***

Question 2. Comment gérer un cinéma

Un cinéma possède s salles de cinéma. Chaque semaine, le cinéma propose une liste de films à voir. Chaque film correspond à un nombre fini de séances. Chaque séance se déroule selon un horaire (intervalle de temps) précis. Les films n'ont pas tous la même durée. On précise qu'il est impossible de changer les horaires des séances.

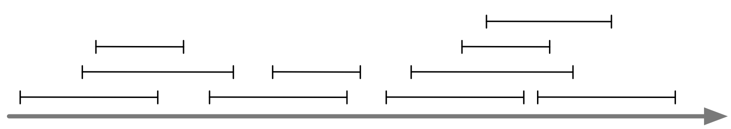

FIGURE 1 – Planning des séances de cinéma

Un étudiant qui a une journée de libre veut voir le maximum de films pendant cette journée (il peut voir un même film plusieurs fois). Ce problème peut être résolu par un algorithme glouton :

Entrée : Un ensemble $\mathcal{I}$ des séances : chaque séance i est caractérisée par l'intervalle (d_i, f_i).
Sortie Un ensemble S des séances.

1. Classer les intervalles par valuations v croissantes : $v(i_1) \leq v(i_2) \leq \cdots \leq v(i_n)$
2. Initialiser la recherche avec $S = \emptyset$;
3. Tant que $\mathcal{I}$ n'est pas vide faire
   (a) Choisir $i \in \mathcal{I}$ qui a la plus petite valuation.
   (b) Ajouter i dans S
   (c) Supprimer toutes les séances de $\mathcal{I}$ qui ne sont pas compatible avec i
4. Retourner S

Maintenant il reste à déterminer la valuation.

Question 2.1 Donner un exemple où l'algorithme ne retourne pas la solution optimale :

1. si la valuation v d'un intervalle est sa date de début (d_i).
2. si la valuation v d'un intervalle est sa durée $(f_i - d_i)$.

Question 2.2 Montrer que l'algorithme est optimal si la valuation v d'un intervalle est sa date de fin (f_i).

Les graphes

Terminologie Un ***graphe non orienté*** G est un couple (V, E) où V est un ensemble de sommets (cercles) et E est un ensemble d'arêtes (trait entre deux cercles). Notons $n = |V|$, $m = |E|$. Les sommets u et v sont dits ***voisins*** s'il y a une arête entre u et v. Nous considérons les notations suivantes.

1. Le ***degré*** de v dans G que l'on notera $d_G(v)$ est le nombre de voisins de v.
2. $\Gamma_G(v)$ est l'ensemble des sommets voisin de v dans G.
3. $G \backslash \{v\}$ est le graphe G privé du sommet de v : son ensemble de sommets est $V \backslash \{v\}$ et son ensemble d'arêtes est $E \backslash \{(u, v) : u \in \Gamma_G(v)\}$.

Question 3. Etant donné un graphe $G = (V, E)$ et un ensemble d'arêtes $M \subseteq E$, évaluer la complexité d'un algorithme qui teste si M est un couplage de G.

Question 4. Dans un graphe orienté sans boucle, à n sommets, un ***trou noir*** est un sommet de degré sortant nul, et de degré entrant $n-1$. Donner un algorithme qui examine $O(n)$ entrées de la matrice d'adjacence pour décider si un graphe sans boucle possède un trou noir. Est-il nécessaire d'examiner $\Omega(n)$ entrées ?

La coupe maximum d'un graphe

Nous allons considérer un graphe non-orienté $G = (V, E)$ de n sommets, ayant une fonction de poids sur les arêtes $w : E \to \mathbb{N}$.

Soit A un sous-ensemble de sommets. Une ***coupe issue de*** A ***noté*** $\mathbf{c(A, B)}$ d'un graphe non-orienté $G = (V, E)$ est un ensemble d'arêtes qui partagent G en deux sous-ensembles disjoints et distincts (A et $B = V \backslash A$). Plus formellement, pour $B = V \backslash A$,

$$c(A, B) = \{(u, v) \in E \text{ tel que } u \in A,\ v \in B\}$$

Cet exercice traite le problème d'optimisation COUPE MAX : nous avons en entrée un graphe non orienté $G = (V, E)$ avec un poids $w(e)$ sur chaque arête e, et nous voulons partager les sommets en deux ensembles A et $B = V \backslash A$ afin que le poids total des arêtes de la coupe $c(A, B)$ soit aussi grand que possible.

Notation : Soit A et B deux ensembles de sommets disjoints. Le ***poids*** de la coupe $c(A, B)$ ***noté*** $\mathbf{w(A, B)}$ est la somme des poids des arêtes de $c(A, B)$. Nous noterons par $\mathcal{W}(\mathbf{A})$, la somme des poids des arêtes dont les deux extrémités sont dans A. Plus formellement,

$$w(A, B) = \sum_{e \in c(A,B)} w(e) = \sum_{u \in A, v \in B, (u,v) \in E} w(u, v) \qquad \text{et } \mathcal{W}(A) = \sum_{u \in A, v \in A, (u,v) \in E} w(u, v)$$

Nous dirons qu'un sommet u est α-***content*** pour la coupe (A, B) dans le graphe G si et seulement si

$$w(A\backslash\{u\}, B\cup\{u\}) \leq \alpha w(A,B) \text{ si } u \in A \text{ ou } w(A\cup\{u\}, B\backslash\{u\}) \leq \alpha w(A,B) \text{ si } u \in B$$

Considérons le graphe biparti complet de 8 sommets (voir la figure 2). Toutes les arêtes de ce graphe ont un poids égal à 1 : i.e. $\forall e \in E$, $w(e) = 1$.

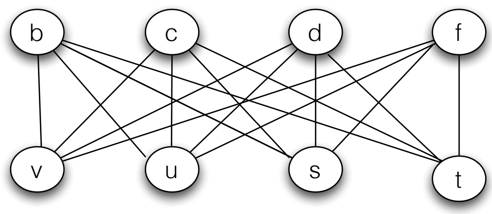

FIGURE 2 – Graphe biparti complet de huit sommets. Toutes ses arêtes ont un poids égal à 1.

Question 5. Compléter les trois tableaux :

A	$B = V\backslash A$	$w(A, V\backslash A)$
$\{b\}$	$\{c,d,f,v,u,s,t\}$	4
$\{b,v\}$		
$\{b,c,v\}$		
$\{b,c,d,f\}$		

A	B	$A\backslash\{b\}$	$w(A\backslash\{b\}, B\cup\{b\})$	$w(A,B)$
$\{b,c,d,f\}$	$V\backslash A$	$\{c,d,f\}$	12	16
$\{b,c,d,v\}$	$V\backslash A$			
$\{b,u,s,t\}$	$V\backslash A$			

A	B	$\mathcal{W}(A)$	$\mathcal{W}(B)$	b est 1-content
$\{b,c,d,f\}$	$V\backslash A$	0	0	oui car ...
$\{b,c,d,v\}$	$V\backslash A$			
$\{b,u,s,t\}$	$V\backslash A$			

Considérons l'algorithme suivant Algorithme MAX-CUT-LOCAL :

Entrée : Un graphe $G = (V, E)$ et une fonction poids $w : E \to \mathbb{N}$

Sortie : Deux sous-ensembles de sommets A et B.

1. Choisir une partition arbitraire de sommets (A, B) de V.
2. Tant qu'il existe un sommet v qui n'est pas α-***content***
   — Si v est dans A, alors $A \leftarrow A\backslash\{v\}$ et $B \leftarrow B \cup \{v\}$

   sinon $B \leftarrow B\backslash\{v\}$ et $A \leftarrow A \cup \{v\}$
3. Retourner les deux ensembles (A, B)

Question 5.1 Exécuter l'algorithme ayant en entrée le graphe de la figure 1 sachant que $\alpha = 1$ et que la partition initiale est la suivante

1. $A = \{b, c, d\}$ et $B = \{v, u, s, t, f\}$
2. $A = \{v, u, b, c\}$ et $B = \{d, f, s, t\}$

Par la suite, la solution par l'algorithme Max-Cut-Local retourne un couple de 2 sous-ensembles de sommets que l'on notera (A, B) tandis qu'une partition correspondant à une coupe maximale sera notée par (A^*, B^*) (c'est-à-dire la coupe $c(A^*, B^*)$ est telle que $w(A^*, B^*)$ soit le grand possible).

Question 5.2 Donner une relation entre $\mathcal{W}(E)$, $\mathcal{W}(A)$, $\mathcal{W}(B)$, $w(A, B)$.

Question 5.3 Montrer que pour tout sommet $u \in A$, on a

$$w(\{u\}, A) \leq w(\{u\}, B) + (\alpha - 1)w(A, B).$$

Indication : u ***est*** α***-content.***

Nous allons supposer pour les prochaines questions que $\alpha = 1$.

Question 5.4 Montrer que pour tout sommet $u \in A$, on a $w(\{u\}, B) \geq \frac{w(\{u\},\Gamma_G(u))}{2}$.

Question 5.5 Montrer que $2w(A, B) \geq \mathcal{W}(E)$ et que $2w(A, B) \geq w(A^*, B^*)$.

Question 5.6 Quelle est la complexité de l'algorithme ? (*indication : évaluer au pire le nombre d'itérations de l'algorithme en fonction du nombre et du poids des arêtes*)

Graphes ayant une fonction de poids quelconque.

Nous considérons un réel arbitraire $\epsilon > 0$. Nous allons supposer par la suite que

$$\alpha = 1 + \frac{2\epsilon}{n}.$$

Question 5.7 Soit (A, B) la solution obtenue par l'algorithme. Montrer que

1. $\sum_{u \in A} w(\{u\}, A) \leq w(A, B) + \frac{|A|2\epsilon}{n} w(A, B)$
2. $\sum_{u \in B} w(\{u\}, B) \leq w(A, B) + \frac{|B|2\epsilon}{n} w(A, B)$
3. $\mathcal{W}(A) + \mathcal{W}(B) \leq (1 + \epsilon)w(A, B)$

Question 5.8 Montrer que $\mathcal{W}(E) \leq (2 + \epsilon)w(A, B)$.

Question 5.9 Montrer que $w(A^*, B^*) \leq (2 + \epsilon)w(A, B)$. Qu'en déduisez-vous ?

Soit (A_0, B_0) la coupe initiale utilisée par l'algorithme. Nous allons considérer que $A_0 = \{v^{max}\}$ et $B_0 = V \backslash A_0$ tel que le sommet v^{max} respecte la condition suivante

$$\forall v \in V, w(\{v^{max}\}, V \backslash \{v^{max}\}) \geq w(\{v\}, V \backslash \{v\}) \tag{1}$$

Question 5.10 Montrer que $w(A_0, B_0) \geq \frac{2}{n}\mathcal{W}(E)$.

Soit (A_t, B_t) la coupe de l'algorithme après l'itération t de la boucle tant que.

Question 5.11 Montrer que $w(A_{t+1}, B_{t+1}) \geq (1 + \frac{2\epsilon}{n})w(A_t, B_t)$

Question 5.12 Donner une borne sur le nombre d'itérations k de l'algorithme pour que le poids de la coupe double ($w(A_{t+k}, B_{t+k}) \geq 2w(A_t, B_t)$). (***Indication :*** $(1 + 1/x)^x \geq 2$ ***pour*** $x \geq 1$)

Question 5.13 Soit K le nombre de l'itérations nécessaires pour que l'algorithme termine. Montrer que $K \geq \frac{|V| log_2 \mathcal{W}(E)}{\epsilon}$. Quelle est la complexité de l'algorithme ?