Complexité et Graphe **2014-2015**
ENSTA

Algorithmes gloutons

Exercice 1 Comment rendre la monnaie.

Nous considérons des pièces de monnaie de 1, 2, et 5 centimes. Notons $N(x)$ le nombre minimun de pièces pour obtenir x centimes.

Question 1.1 Quelle est la valeur de $N(0)$, $N(1)$, $N(2)$, $N(3)$, $N(4)$, $N(5)$?

Question 1.2 Donner un algorithme qui calcule $N(x)$ et sa complexité en terme d'opérations.

Question 1.3 Appliquer cet algorithme qui calcule $N(14)$ en considérant maintenant qu'on dispose des pièces de monaies de 1, 7 et 10 centimes. Que constatez-vous ?

Exercice 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.

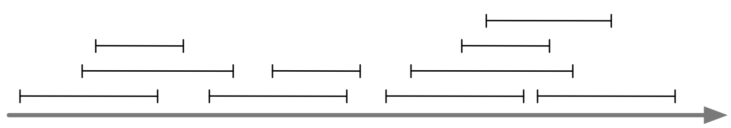

Figure 1 – Planning des séances de cinéma

Question 2.1 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. 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 $(f_i - d_i)$.

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

Question 2.2 La personne qui gère l'affectation des salles doit affecter à chaque séance une salle. Son problème est de comprendre si le nombre de salles est suffisant.

Proposer un algorithme polynomial. Justifier. *Indication : on pourra considérer les séances dans l'ordre de leur date de début.*

Exercice 3 La théorie des matroïdes permet de comprendre si un algorithme glouton est optimal pour un problème. Voici la définition d'un matroïde.

— $(\mathcal{X}, \mathcal{F})$ est un matroïde si $\mathcal{X}$ est un ensemble de n éléments, $\mathcal{F}$ est une famille de parties de $\mathcal{X}$ $(\mathcal{F} \subset \mathcal{P}(\mathcal{X}))$ vérifiant
  1. ≪ **Hérédité** ≫ : $X \in \mathcal{F}$ implique $\forall Y \subset X$, $Y \subset \mathcal{F}$
  2. ≪ **Echange** ≫ : $(A, B \in \mathcal{F}, |A| < |B|)$ implique $\exists x \in B - A$ tel que $A \cup \{x\} \in \mathcal{F}$.

— Les éléments de $\mathcal{F}$ sont appelés ≪ *indépendants* ≫.

Question 3.1 Montrer que les forêts d'un graphe correspondent à un matroïde. *(indication : une forêt qui contient x arbres possèdent $n - x$ arêtes.)*

Un indépendant est dit ≪ *maximal* ≫ s'il est maximal au sens de l'inclusion.

Question 3.2 Montrer que tous les indépendants maximaux ont le même cardinal.

Considérons les **matroïdes pondérés**

— On ajoute une fonction de poids $w : x \in \mathcal{X} \mapsto w(x) \in \mathbb{R}^{\geq 0}$.

— On pose
$$w(X) = \sum_{x \in X} w(x).$$

Question 3.3 Etant donné un matroïde pondéré, considérons l'algorithme glouton qui construit un indépendant de poids maximal (vu en cours)

1. Trier les éléments de $\mathcal{X}$ par poids décroissants : $w(s_1) \geq w(s_2) \geq \cdots \geq w(s_n)$.
2. $A := \emptyset$.
3. Pour i=1 à n faire
   (a) Si $A \cup \{s_i\} \in \mathcal{F}$ alors $A := A \cup \{s_i\}$
4. retourner A

Prouver la validité de votre algorithme.

Exercice 4 Ordonnancement de tâches

Le problème de l'ordonnancement de tâches sur un seul processeur se compose

— d'un ensemble de tâches $\mathcal{X} = \{1, \ldots, n\}$ de durée 1,

— avec des dates limites $d_1, \ldots, d_n$: la tâche i doit finir avant la date d_i, sinon on doit payer la pénalité w_i.

L'objectif est de trouver un ordonnancement des tâches pour minimiser la somme des pénalités.

Afin de résoudre ce problème, on utilise les définitions suivantes :

— Un ensemble A de tâches est dit *indépendant* s'il est possible de les ordonnancer de sorte que personne ne soit en retard.

— On note $N_t(A)$ le nombre de tâches de l'ensemble A dont la date limite est $\leq t$.

Question 4.1 Montrer que les 3 propriétés suivantes sont équivalentes

1. A est indépendant ;
2. Pour $t = 1, \ldots, n$, on a $N_t(A) \leq t$;
3. Si les tâches de A sont ordonnancées dans l'ordre croissant de leur dates limites, alors aucune tâche n'est en retard.

Question 4.2 Montrer que $(\mathcal{X}, \mathcal{F})$, où $\mathcal{X}$ est l'ensemble des tâches et $\mathcal{F}$ la famille des sous-ensembles de tâches indépendantes, vérifie les propriétés suivantes :

1. *Hérédité* : $X \in \mathcal{F}$ implique $\forall Y \subset X$, $Y \subset \mathcal{F}$
2. *Échange* : $(A, B \in \mathcal{F}, |A| < |B|)$ implique $\exists x \in B - A$ tel que $A \cup \{x\} \in \mathcal{F}$.

Question 4.3 Déterminer un algorithme glouton qui fournit un ensemble indépendant A optimal.

Question 4.4 Appliquer cet algoritme à l'exemple suivant composé de 7 tâches :

i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	7	6	5	4	3	2	1

Exercice 5 Coloriage des graphes planaires

Un graphe est *planaire* si on peut le dessiner dans le plan sans que les arêtes ne se croisent.

Soit $G = (V, E)$ un graphe planaire non vide *connexe*. Notons $n = |V|$, $a = |E|$. Considérons un dessin de G dans le plan (sans que les arêtes ne se croisent). Soit f le nombre de régions du plan délimitées par les arêtes de G.

Question 5.1 Montrer que les graphes planaires respectent la relation d'Euler $n - a + f = 2$. (*Indication : raisonner par récurrence sur le nombre de faces*)

Question 5.2 On suppose $n \geqslant 3$. Montrer que $3f \leq 2a$.

Question 5.3 Montrer qu'il existe un sommet de degré au plus 5 dans le graphe planaire G.

Question 5.4 Concevoir un algorithme de 6-coloration des graphes planaires en temps polynomial.

Remarque : En fait, tout graphe planaire peut être colorer avec 4 couleurs. Ce résultat est connu sous le nom de théorème des quatre couleurs. Il a été démontré en 1976 par Appel et Haken.