Algorithmes gloutons

Le principe de l'algorithme glouton : faire toujours un choix localement optimal dans l'espoir que ce choix mènera à une solution globalement optimale.

1 Égypte

On appelle fraction égyptienne une fraction de la forme $\frac{1}{n}$ avec $n \in \mathbb{N}^*$.

1. Soient a et b deux entiers (premiers entre eux) tels que $a < b$. Donner l'expression de la fraction égyptienne la plus grande parmi les fractions égyptiennes strictement plus petites que $\frac{a}{b}$ et l'expression de leur différence.
2. On considère l'algorithme suivant :

   Xcas

   ```
   egypte(a,b):={
   local c,d;
   c:=a;
   a:=numer(c/b);
   b:=denom(c/b);
   si a==1 alors return  b  ;
   sinon
   c:=a-irem(b,a);
   d:=iquo(b,a)+1;
   return (d,egypte(c,b*d));
   fsi;
   }:;
   ```

   dans lequel l'entrée $(a; b)$ est un couple d'entiers avec $a < b$.
   Écrire une version itérative de l'algorithme.
3. Quelle est la sortie de l'algorithme avec l'entrée $(a, b) = (2, 2n+1)$ où n est un entier naturel non nul ?
4. Quel est le rôle de cet algorithme ? Démontrer.
5. L'algorithme glouton proposé donne-t-il une décomposition en somme de fractions égyptiennes avec le minimum de termes possibles ?

2 Les épreuves dans le gymnase

Dans un gymnase doivent se dérouler une série d'épreuves. Les épreuves ne sont pas seulement caractérisées par leurs durées : chaque épreuve est caractérisée par une date de début d_i et une date de fin f_i.

On souhaite "caser" le plus possible d'épreuves, deux épreuves ne pouvant avoir lieu en même temps (leurs intervalles de temps doivent être disjoints).

Glouton 1 – On trie les épreuves par durée croissante, on choisit la plus courte, puis la plus courte parmi celles qui lui sont compatibles, puis ...Ce choix mène-t-il au déroulement d'un nombre d'épreuves maximal ?

Glouton 2 – On trie les événements par dates de commencement croissantes et on gloutonne : on choisit l'événement commençant le plus tôt, puis le plus tôt parmi les événements compatibles ...Même question.

Glouton 3 – On trie cette fois les événements par nombre d'intersections croissant : on choisit d'abord celui qui intersecte le moins d'événements, puis ...Même question.

Glouton 4 – On trie les événements par dates de fin croissantes et on gloutonne : on choisit l'épreuve se terminant au plus tôt, puis l'épreuve se terminant au plus tôt parmi celles qui sont compatibles à la première...Même question.

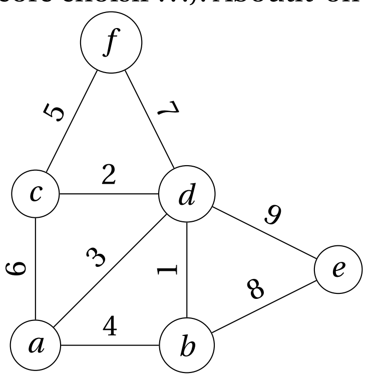
3 Monnaie

1. On dispose de pièces de monnaie (sans limite d'effectifs) de 18 chloubis , 7 chloubis et 1 chloubi. On cherche la façon de payer n chloubis (n entier naturel) en utilisant le moins possibles de pièces. On gloutonne ainsi : on utilise des pièces de 18 chloubis tant qu'on peut, puis des pièces de 7 tant qu'on peut, puis des pièces de 1. Cette gloutonnerie donnera-t-elle une réponse optimale ?
2. La même gloutonnerie conduit-elle à un nombre minimal de pièces lorsque les pièces sont des pièces 10, 5, 2 et 1 ?
3. Soit $p \in \mathbb{N}$, $p \geqslant 2$. La même gloutonnerie conduit-elle à un nombre minimal de pièces lorsque les pièces sont des pièces de 1, p, p^2, p^3, ..., p^k ?

4 Deux algorithmes sur des graphes

4.1 Couplage

Soit G un graphe, on appelle couplage tout ensemble d'arêtes tel que deux arêtes quelconques de cet ensemble ne sont jamais incidentes à un sommet commun. Les arêtes du graphe ci-dessous sont pondérées. On aimerait déterminer un couplage de poids maximal. Appliquer le principe glouton (c'est à dire : choisir l'arête de poids maximal, puis l'arête de poids maximal parmi les arêtes que l'on peut encore choisir ...). Aboutit-on à un couplage de poids maximal ?



4.2 Arbre de poids maximal

Algorithme de Kruskal.
Un arbre dans un graphe G est un sous-graphe qui ne possède aucun cycle. Les arêtes de l' arbre ci-dessous sont pondérées. Le choix glouton (on choisit l'arête de poids maximal, puis l'arête de poids maximal parmi celles qui peuvent encore être choisies...) mène-t-il à un arbre de poids maximal ?
On admettra : « Tous les arbres couvrants d'un graphe connexe G ont le même nombre d'arêtes (à savoir $n-1$ où n est le nombre de sommets). »

4.3 Matroïde

1. Les couplages d'un graphe constituent une famille d'ensembles « héréditaire » : si un ensemble H d'arêtes est un couplage du graphe G alors tout ensemble H' d'arêtes contenu dans H est aussi un couplage du graphe G.

   Les ensembles d'arêtes d'un graphe G engendrant un graphe sans cycle vérifient également cette propriété d'hérédité : si un ensemble H d'arêtes engendre un graphe sans cycle alors tout ensemble H' d'arêtes contenu dans H engendre aussi un graphe sans cycle.

2. Formulation de l'algorithme glouton pour un couple $(E, \mathscr{I})$ où E est un ensemble fini d'éléments (par exemple : l'ensemble des arêtes d'un graphe) et $\mathscr{I}$ une famille de parties de E héréditaire (par exemple, la famille des couplages ou la famille des graphes sans cycles d'un graphe). Les éléments de $\mathscr{I}$ seront nommés parties indépendantes.

Entrée	le couple $(E, \mathscr{I})$ et une fonction poids w (à valeurs positives) définie sur E.

Traitement	$J := \varnothing$, $A := E$
	TantQue $A \neq \varnothing$
	    $e :=$ élément de A de poids maximal
	    $A := A - e$
	    Si $J + e$ est une partie indépendante alors $J := J + e$
	FinTantQue

Sortie	J

   L'algorithme se termine toujours (puisque E est fini). Sous certaines conditions (voir ci-dessous), il donnera une partie indépendante de poids maximal.

3. La famille des ensembles d'arêtes d'un graphe engendrant un graphe sans cycle vérifie la propriété d'échange (on ne le démontrera pas ici) : « si H et H' sont des ensembles d'arêtes du graphe G engendrant un graphe sans cycle et si $|H'| < |H|$ alors il existe un élément e de H tel que $H' + e$ engendre un graphe sans cycle. »

   De façon plus générale, un système $(E, \mathscr{I})$ héréditaire sera appelé matroïde s'il vérifie la propriété d'échange : « si H et H' sont des parties indépendantes et si $|H'| < |H|$ alors il existe un élément e de H tel que $H' + e$ est une partie indépendante. »

   (a) Vérifier que la famille des couplages d'un graphe ne possède pas la propriété d'échange.

   (b) Vérifier que pour un système héréditaire qui n'est pas un matroïde l'algorithme glouton énoncé ci-dessus peut ne pas mener à une partie indépendante de poids maximal.

(c) Vérifier que, pour un matroïde, l'algorithme glouton ci-dessus mène à une partie indépendante de poids maximal (et en particulier, l'algorithme de Kruskal donne un arbre couvrant de poids maximal).

5 Le sac à dos

5.1 Le sac à dos du cambrioleur

1. Un voleur dévalisant un magasin trouve n objets. L'objet numéro i vaut v_i euros et pèse w_i kg. Le voleur veut que son butin ait la plus grande valeur (en euros) possible mais ne peut pas emporter plus de W kg dans son sac à dos.

   Glouton 1 – Le résultat est-il optimal en choisissant l'objet le plus cher parmi ceux qui peuvent tenir dans le sac, puis le plus cher parmi ceux qui peuvent encore tenir ...

   Glouton 2 – Le résultat est-il optimal en choisissant d'abord les objets de plus grand prix au kg ?
2. Les objets sont maintenant des quantités fractionnables (par exemple 10 kg d'une certaine poudre). L'algorithme glouton consistant à charger dans le sac à dos la plus grande quantité du produit le plus cher au kg, puis la plus grande quantité du produit le plus cher au kg parmi ceux qui restent ...donne-t-il une solution optimale ?

5.2 Le sac à dos de l'espion

La présentation du sac à dos dans cette section a été utilisée dans un algorithme de chiffrement (Merkle-Hellman) pour lequel on pourra trouver le principe détaillé ici :
`http://www.apprendre-en-ligne.net/crypto/menu/index.html`
On dispose d'une suite finie d'entiers s_1, s_2, ..., s_k super-croissante, c'est à dire telle que $s_j > \sum_{\ell=1}^{j-1} s_\ell$ ($2 \leqslant j \leqslant k$). On se donne un entier $C > 0$ et on cherche si la somme de certains éléments s_j est égale à C, c'est à dire si l'on peut trouver $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k) \in \{0;1\}^k$ tel que $\sum_{j=1}^{k} \varepsilon_j s_j = C$.
Montrer que l'algorithme glouton ci-dessous résout le problème :

Xcas

```
sacados(liste,C):={
// liste : séquence super croissante d'entiers
// C : somme à atteindre
local j, longueur_liste,solu,s;
longueur_liste:=size(liste);
solu:=seq[];
pour j de longueur_liste-1 jusque 0 pas -1 faire
s:=liste[j];
si s<=C alors solu:=solu,s; C:=C-s;fsi;
fpour;
si C==0 alors return solu;
sinon return "pas de solution";
fsi;}:;
```

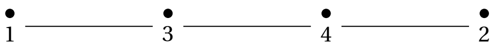
sacados([13,45,183,315,801],511) renvoie par exemple 315,183,13.

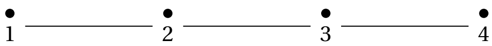
6 Coloration des sommets d'un graphe

On cherche à obtenir une coloration des sommets d'un graphe qui satisfasse à la contrainte suivante : deux sommets voisins n'ont jamais la même couleur. Une question se pose : quel est le plus petit nombre de couleurs permettant de colorier les sommets d'un graphe sous cette contrainte ? (ce plus petit nombre est appelé nombre chromatique du graphe).
On considère l'algorithme suivant :

Donnée – un graphe G et des couleurs 1,2,3,4...Les sommets de G sont numérotés de 1 à n ($s_1, s_2, \ldots, s_n$).
Processus – pour i allant de 1 à n, affecter au sommet s_i la plus petite couleur non déjà affectée à ses voisins déjà coloriés (c'est-à-dire la plus petite couleur non déjà affectée à ceux des sommets $s_1, s_2, \ldots, s_{i-1}$ qui lui sont adjacents). En d'autres termes, on gloutonne : on prend localement le plus petit nombre possible.
Sortie – Une coloration valide du graphe G. Mais le nombre de couleurs utilisées est-il minimal ?

6.1 L'algorithme ne fournit pas nécessairement une coloration optimale

1. Appliquer cet algorithme au graphe ci-dessous avec les deux numérotations des sommets proposées :
   (a)
   

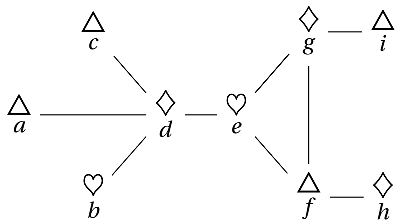
   (b)
   

2. Cet algorithme donne-t-il le nombre chromatique du graphe ?

6.2 Nombre maximal de couleurs utilisées par l'algorithme

Montrer que cet algorithme donnera toujours une coloration utilisant au plus $\Delta(G) + 1$ couleurs (où Δ désigne le degré maximal des sommets).

6.3 L'algorithme peut donner une coloration optimale

1. Montrer que la coloration du graphe ci-dessous est optimale mais qu'elle ne peut pas être obtenue par l'algorithme.
   

2. Établir qu'il existe toujours une numérotation initiale de G telle que l'application de l'algorithme donne une coloration optimale (c'est à dire avec $\chi(G)$ couleurs).

6.4 Une très mauvaise coloration ?

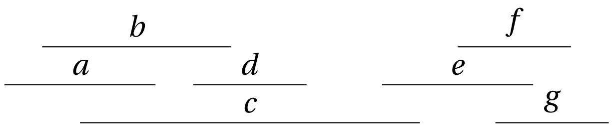
L'algorithme présenté ne donne pas nécessairement une coloration optimale. Mais une très mauvaise coloration (du point de vue du nombre de couleurs utilisées) est-elle possible ?
Soit $k \geqslant 2$ un entier. Construire un graphe G de nombre chromatique 2 et une numérotation des sommets de ce graphe G telle que l'application de l'algorithme précédent donne une coloration avec k couleurs.

6.5 Graphe d'intervalles

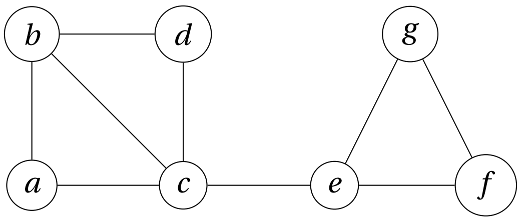
On reprend le problème du gymnase. On cherche maintenant le nombre minimal de gymnases permettant le déroulement de tous les événements.

Glouton 1 – Les dates de fin au plus tôt ayant permis le déroulement d'un nombre optimal d'épreuves avec un seul gymnase, on essaie de "remplir" le gymnase 1 au maximum avec ce principe, puis on passe à un gymnase 2, puis ...Le résultat sera-t-il optimal ?

Glouton 2 – On représente par exemple les intervalles de temps :



par le graphe :



Montrer qu'une coloration du graphe par l'algorithme glouton décrit plus haut utilise un nombre de couleurs égal au nombre chromatique avec une numérotation des sommets correspondant à l'ordre des extrémités gauche des intervalles.

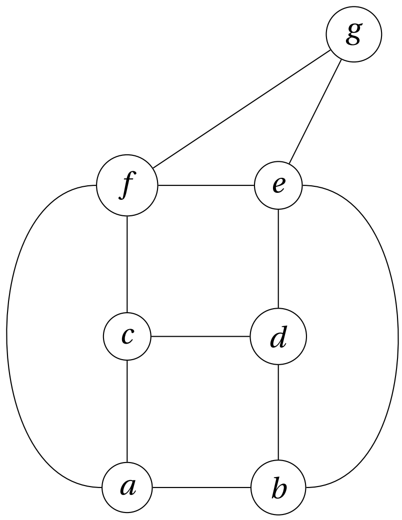
6.6 Algorithme de Brelaz

On définit, à toute étape de l'algorithme, le degré-couleur d'un sommet comme le nombre de couleurs déjà utilisées pour ses voisins.

Donnée – un graphe G et des couleurs 1,2,3,4...(les degrés-couleur sont initialisés à 0).
Processus – Prendre parmi les sommets de degré-couleur maximal un sommet de degré maximal, lui attribuer la plus petite couleur possible. Mettre à jour les degrés-couleur.
Sortie – Une coloration valide du graphe G. Mais le nombre de couleurs utilisées est-il minimal ?

Exercice

1. Appliquer l'algorithme au graphe ci-dessous :



2. Montrer que la coloration obtenue n'est pas nécessairement optimale (c'est à dire : peut demander un nombre de couleurs strictement supérieur au nombre chromatique).
3. Montrer que l'algorithme de Brelaz colore les graphes bipartis en deux couleurs.