Algorithmes gloutons

1 L'affaire est dans le sac : un premier problème d'optimisation

Un cambrioleur possède un sac à dos d'une contenance maximum de 30Kg. Au cours d'un de ses cambriolages, il a la possibilité de dérober 4 objets A, B, C et D. Voici un tableau qui résume les caractéristiques de ces objets :

caractéristiques des objets objet	A	B	C	D
masse en Kg	13	12	8	10
valeur marchande en €	650	400	352	350

Exercice n° 1

Déterminez les objets que le cambrioleur aura intérêt à dérober, sachant que :
— tous les objets dérobés devront tenir dans le sac à dos (30 Kg maxi)
— le cambrioleur cherche à obtenir un gain maximum.

Ce genre de problème est un grand classique en informatique, on parle de problème d'optimisation. Dans le problème du sac à dos, on peut choisir A+B ou A+C ou B+C+D... toutes les combinaisons sont possibles à partir du moment où la masse totale ne dépasse pas 30 Kg, mais on ne cherche pas n'importe quelle solution, on cherche une solution dite optimale : dans notre exemple on cherche le plus grand gain possible.

Il existe différentes méthodes algorithmiques permettant de trouver une solution optimale à un problème d'optimisation : il peut, en effet, être intéressant "d'automatiser" la résolution des problèmes d'optimisation à l'aide d'algorithme.

Peut-on trouver un algorithme qui trouve une solution optimale au problème du sac à doc ?

En apparence, la solution la plus simple dans le cas du sac à dos serait d'écrire un algorithme qui teste toutes les combinaisons d'objets possibles et qui retient les solutions qui offrent un gain maximum. Dans notre cas précis, avec seulement 4 objets, cette solution pourrait être envisagée, mais avec un plus grand nombre d'objets, le temps de calculs, même pour un ordinateur très puissant, deviendrait trop important. En effet l'algorithme qui testerait toutes les combinaisons possibles aurait une complexité en temps en $O(a^n)$ avec a une constante et n le nombre d'objets. On parle d'une complexité exponentielle. Les algorithmes à **complexité exponentielle** ne sont pas efficaces pour résoudre des problèmes, le temps de calcul devient beaucoup trop important quand n devient très grand.

À la place de cette méthode "je teste toutes les possibilités", il est possible d'utiliser une méthode dite gloutonne (greedy en anglais). **Qu'est-ce qu'une méthode gloutonne ?**

La résolution d'un problème d'optimisation se fait généralement par étapes : à chaque étape on doit faire un choix. Par exemple, dans le problème du sac à dos, nous ajoutons les objets un par un, chaque ajout d'un objet constitue une étape : à chaque étape on doit choisir un objet à mettre dans le sac. Le principe de la méthode gloutonne est que, à chaque étape de la résolution du problème, faire le choix qui semble le plus pertinent sur le moment, avec l'espoir qu'au bout du compte, cela nous conduira vers une solution optimale du problème à résoudre. Autrement dit, on fait des choix localement optimaux dans l'espoir que ces choix mèneront à une solution globalement optimale.

On procède de façon séquentielle, en faisant à chaque étape le choix qui semble localement le meilleur. On ne revient jamais en arrière.
Il s'agit d'une progression descendante, à chaque étape on fait un choix puis on résout un problème plus petit issu de ce choix.

Appliquons une méthode gloutonne à la résolution du problème du sac à dos :

Méthode gloutonne 1 :

Méthode gloutonne 2 :

Sachant que l'on cherche à maximiser le gain, commençons par établir un tableau nous donnant la "valeur massique" (valeur par unité de masse) de chaque objet (pour chaque objet on divise sa valeur par sa masse) :

valeur massique des objets objet	A	B	C	D
valeur massique en €/Kg	50	33	44	35

On classe ensuite les objets par ordre décroissant de valeur massique : A - C - B -D
Enfin, on remplit le sac en prenant les objets dans l'ordre et en s'arrêtant dès que la masse limite est atteinte. C'est ici que ce fait "le choix glouton", à chaque étape, on prend l'objet ayant le rapport "valeur-masse" le plus intéressant au vu des objectifs :

- 1ère étape : A (13 Kg)
- 2ème étape : C (13+8=21 Kg)
- 3ème étape : B (13+8+12=33 Kg) -¿ impossible, on dépasse les 30 Kg.

Solution gloutonne 2 : Le sac est donc composé de 2 objets : A et C pour un montant total de 1000 €et une masse totale de 21 Kg.

Cette méthode gloutonne peut être "automatisée", il est donc possible d'écrire un algorithme glouton (un algorithme qui est basé sur une méthode gloutonne) afin de trouver une solution au problème du sac à dos avec n'importe quelles valeurs (nombre d'objets, masse des objets, valeur des objets, masse maximum).

La solution trouvée ci-dessus est-elle optimale ?

On constate rapidement que la réponse est non, car

Plus généralement, il est important de bien comprendre qu'un algorithme glouton ne donne pas forcement une solution optimale.

2 Le problème du rendu de monnaie

Nous sommes des commerçants, nous avons à notre disposition un nombre illimité de pièces de :

— 1 centime
— 2 centimes
— 5 centimes
— 10 centimes
— 20 centimes
— 50 centimes
— 1 €
— 2 €

Nous devons rendre la monnaie à un client à l'aide de ces pièces. La contrainte est d'utiliser le moins de pièces possible.

Exercice n° 2

---

1. Trouvez une méthode gloutonne permettant d'effectuer un rendu de monnaie (en utilisant le moins possible de pièces).

2. Vous devez rendre la somme de $2,63$ €, appliquez la méthode que vous venez de mettre au point. Combien de pièces avez-vous utilisées ?

Exercice n° 3

À partir de la méthode gloutonne que vous avez élaborée ci-dessus, écrivez un algorithme glouton qui permettra de déterminer le nombre minimal de pièces à utiliser pour une somme donnée.

Remarque : Avec notre système de monnaie actuel, l'algorithme glouton de l'exercice précédent est optimal.

Exercice n° 4

Imaginons un système monétaire constitué uniquement de pièces de 4€, 3€et 1€. L'algorithme glouton de l'exercice précédent donne t-il une solution optimale ?

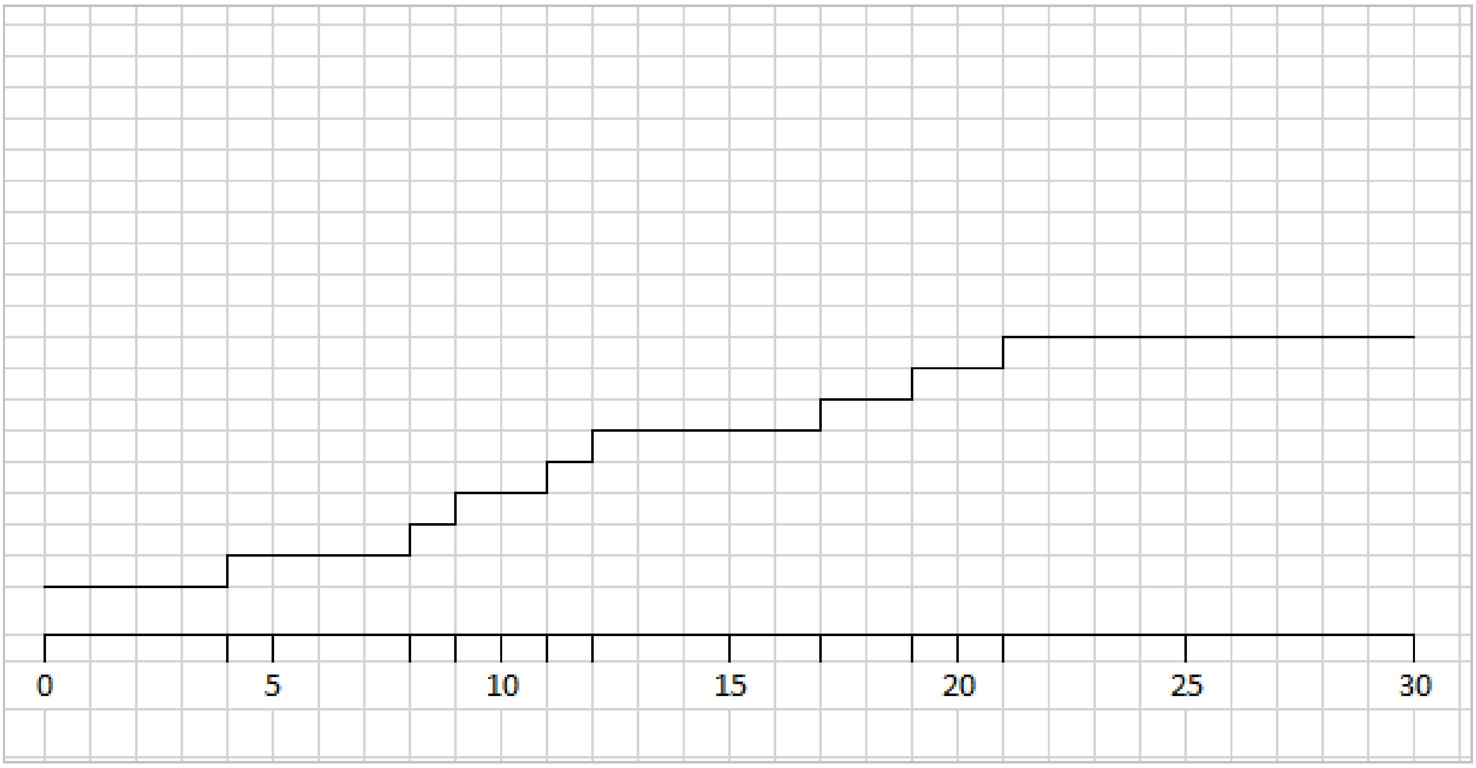
3 C'est Versailles !

Il y a une longue allée dans le jardin d'Alice avec n marches d'escalier placées de manière irrégulière.

Elle souhaite donc installer des lanternes sur le chemin pour éviter que ses invités ne choient s'ils s'aventurent au clair de lune. Alice veut faire en sorte que chaque marche soit éclairée. Elle a repéré un modèle de lanternes de portée p (c'est-à-dire qu'une lanterne placée en position x éclaire une zone comprise entre $x - p$ et $x + p$) qui lui plaît bien mais qui est un peu cher.

Exercice n° 5

1. Positions des marches : 4, 8, 9, 11, 12, 17, 19, 21
   Portée des lampadaires : L = 3 un lampadaire éclaire les 3 carreaux situés à sa gauche **et** les 3 carreaux situés à sa droite.



   Positions des lampadaires :
   Nombre minimum de lampadaires nécessaires :

2. Proposer un algorithme glouton pour aider Alice à placer un nombre minimum de lanternes permettant d'éclairer chaque marche.