T.P. 06

Algorithmes gloutons

■ ■ ■

Les algorithmes servant à résoudre les problèmes d'optimisation parcourent en général une série d'étapes, au cours desquelles ils sont confrontés à un ensemble de possibilités. Un algorithme glouton fait toujours le choix qui semble le meilleur *sur le moment*, dans l'espoir qu'il mènera à la solution optimale *globalement*.
Chaque partie de ce sujet constitue un exercice indépendant.

Partie I : Le voleur intelligent

Un cambrioleur entre par effraction dans une maison et désire emporter quelques-uns des objets de valeur qui s'y trouvent. Il n'est capable de porter que x kilos : il lui faudra donc choisir entre les différents objets, suivant leur valeur (il veut bien entendu amasser le plus gros magot possible...).
On suppose dans un premier temps que les objets sont des matières fractionnelles (on peut en prendre n'importe quelle quantité, c'est le cas d'un liquide ou d'une poudre). Il y a n matières différentes, numérotées de 0 à $n-1$, la i-ème ayant un prix $p.(i)$ par kilo. La quantité disponible de cette matière est $q.(i)$. On suppose que tous les prix sont différents deux à deux.

Exercice 1. Proposez un algorithme qui donne un choix optimal pour le voleur. Ce choix est-il unique ? Programmez une fonction voleur en Caml qui reprenne cet algorithme (vous pourrez supposer que le tableau p est trié).

value voleur : $int\ vect \rightarrow int\ vect \rightarrow int \rightarrow int\ vect$

On suppose maintenant que les objets sont non fractionnables (c'est le cas d'une chaise ou d'un téléviseur). Le i-ème objet vaut un prix $p.(i)$ et pèse un poids $q.(i)$.

Exercice 2. Proposez une méthode dérivée de la question 1 (sans la programmer en Caml). Donne-t-elle un choix optimal ?

Partie II : Réservation d'une salle

Un organisateur de tournois sportifs désire utiliser au mieux le gymnase local lors de la journée « portes ouvertes ». Il y a n événements numérotés de 0 à $n-1$. L'événement numéro i commence à l'heure d_i et finit à l'heure f_i. Autrement dit, l'événement i recquiert le gymnase durant l'intervalle de temps $[d_i; f_i[$. Le problème est de planifier le nombre maximal d'événements parmi les n dans le gymnase.

Exercice 3. Indiquez comment modéliser la situation. Proposez un algorithme et écrivez une fonction Caml qui résout le problème. On pourra supposer que $f_0 \leqslant f_1 \leqslant \ldots \leqslant f_{n-1}$.
L'idée gloutonne est sélectionner un à un des événements en prenant à chaque fois celui qui termine le plus tôt parmi ceux qui sont encore possibles. Le plus simple est probablement d'écrire une fonction récursive...

value salle : $int\ vect \rightarrow int\ vect \rightarrow int\ list$

Exercice 4. Pouvez-vous adapter votre algorithme de manière à optimiser la durée d'utilisation du gymnase ?

Partie III : L'angoisse de la panne sèche

Une route comporte n stations services, numérotées dans l'ordre du parcours, de 0 à $n-1$. La première est à une distance $d.(0)$ du départ, la deuxième est à une distance $d.(1)$ de la première,

la troisième à une distance $d.(2)$ de la deuxième, etc., la fin de la route est à une distance $d.(n)$ de la n-ième et dernière station service.
Un automobiliste prend le départ de la route avec une voiture dont le réservoir d'essence est plein. Sa voiture est capable de parcourir une distance r (mais pas plus !) avec un plein.

Exercice 5. Donnez une condition nécessaire et suffisante pour que l'automobiliste puisse effectuer le parcours. On la supposera réalisée par la suite.

Exercice 6. L'automobiliste désire faire le plein le moins souvent possible. Écrivez une fonction Caml rapide qui détermine à quelles stations-service il doit s'arrêter.
Une méthode consiste à toujours aller le plus loin possible sans faire le plein, c'est-à-dire à faire systématiquement le plein à la dernière station avant la panne sèche...
value rapide : $int\ vect \to int \to int\ list$
Maintenant, l'automobiliste part avec un réservoir vide. Il doit au départ acheter de l'essence (autant qu'il veut dans la limite de la contenance r de son réservoir) au prix de $e.(0)$ euros par litre. Par la suite, à la station numéro i, l'essence coûte $e.(i)$ euros par litre.

Exercice 7. La voiture consomme exactement un litre de carburant par unité de distance. Écrivez une fonction econome qui indique à quelles stations l'automobiliste doit s'arrêter et combien de litres il doit prendre à chaque fois, afin que son trajet lui coûte le moins cher possible.
value econome : $int\ vect \to int \to int\ list$

Partie IV : Réservation SNCF

On suppose que n personnes veulent voyager en train un jour donné. La personne i veut prendre le train $p.(i)$, où p est un tableau d'entiers. Les trains sont numérotés de 0 à $k-1$ et partent dans l'ordre de leur numéro. Chaque train peut contenir au plus c personnes.

Exercice 8. Écrivez une fonction possible qui teste si tout le monde peut prendre le train de son choix. Vous veillerez à ce que votre fonction ne modifie pas le tableau passé en argument.
value possible : $int \to int\ vect \to int \to bool$
On suppose maintenant que la personne i, si elle ne peut pas prendre le train $p.(i)$ parce qu'il est complet, accepte de prendre un des trains suivants (s'il y en a un).

Exercice 9. Proposez et programmez un algorithme sncf pour affecter chaque personne à un train lorsque c'est possible.
value sncf : $int \to int\ vect \to int \to int\ vect$