Cours NSI 1^{re}

Algorithmes Glouton

1 Introduction

Les algorithmes gloutons regroupent différents algorithmes servant en général à résoudre des problèmes d'optimisations.
Un problème d'optimisation est un problème qui consiste à minimiser ou maximiser une fonction sur un ensemble.
Des exemples de problèmes d'optimisation sont:

- De la répartition de tâches (par exemple la création d'emplois du temps).
- Le problème du voyageur.

Ainsi que 2 problèmes que nous allons étudier:

- Le problème du rendu de monnaie.
- Le problème du sac à dos.

Les algorithmes gloutons sont une approche a différents problèmes de ce type et apportent une solution exacte ou approchée à ces problèmes.
Ils fonctionnent en faisant, à chaque étape, le choix optimal. Selon les problèmes cela peut mener à un résultat optimal ou approximatif au problème.

2 Le glouton avec un résultat optimal: Le rendu de monnaie

Suite à un achat en espèces, il arrive que l'on nous rende de la monnaie.
Considérons pour simplifier le problème, que l'on puisse nous rendre de la monnaie uniquement avec les devises (pièces ou billet, peut importe): 1,2,5,10,20 ou 50€.
Le problème du **rendu de monnaie** consiste à déterminer la solution ou l'on rends le nombre minimal de pièces.

Exercice 1

Question 1:

Pour rendre 49€, comment feriez vous ? .....

Même question pour 7€. .....

Question 2:

Proposez une stratégie (optimale ou non) au problème de rendu de monnaie.

.....

.....

.....

Algorithme glouton pour le problème de rendu de monnaie

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 2

L'algorithme glouton que nous avons écrit ci-dessus serait-il toujours optimal dans le cas ou on aurait des pièces de 7€ ?

.....

3 Un problème plus compliqué: Le sac à dos

Le problème du sac à dos est un autre problème d'optimisation. Il modélise une situation similaire à celle du remplissage d'un sac à dos (d'ou le nom du problème) qui ne peut **supporter qu'un certain poids**. En plus de ce sac nous avons des objets qui ont un **poids** et une **valeur**.
L'objectif du problème est de maximiser la valeur totale des objets dans le sac, sans dépasser le poids limite.
Ce problème peut se résoudre par force brute, c'est-à-dire en testant tous les cas possibles. Mais ce type de résolution présente un problème d'efficacité. Son coût en fonction du nombre d'objets disponibles croît de manière exponentielle.

4 Un point d'histoire

Le problème du sac à dos est l'un des 21 problèmes NP-complets exposés par un article écrit un 1972 par Richard Karp. Ce problème est fortement étudier depuis le milieu du 20^e siècle.

Exercice 1

Un critère pour faire un choix optimal local pourrait être "Prendre à chaque fois l'objet le moins lourd".
Proposez deux autres critères:

. . . . . .

Exercice 2

Proposez des exemples ou la stratégie gloutonne utilisant les différents critères n'est pas optimale.

Exercice 3

Écrire un algorithme glouton pour chaque stratégie avec les "objets" proposée sur la page suivante.
Le poids maximal que le sac accepte est de 100.
Les objets sont représentés par un tableau contenant 100 objets qui sont eux-même représentés par un tableau de 2 éléments:

- le premier élément est le poids.
- le deuxième élément est la valeur.

objets = [[30, 16.0], [15, 7.0], [10, 5.0], [15, 22.0], [26, 21.8], [5, 3.0], [13, 19.2], [12, 17.8], [21, 13.6], [25, 8.5], [19, 12.4], [21, 17.8], [30, 22.0], [28, 17.8], [29, 41.6], [1, 0.6], [19, 21.9], [72, 55.0], [1, 0.6], [14, 6.6], [27, 38.8], [75, 55.0], [1, 0.5], [72, 51.0], [28, 23.4], [5, 4.5], [30, 31.0], [1, 0.6], [11, 8.7], [19, 20.0], [30, 34.0], [7, 8.7], [17, 18.0], [15, 14.5], [11, 9.8], [22, 16.4], [19, 25.7], [15, 7.0], [4, 5.8], [5, 5.5], [3, 2.8], [2, 0.7], [2, 0.5], [16, 15.4], [5, 5.0], [1, 0.6], [6, 3.4], [73, 51.0], [13, 11.4], [9, 12.7], [24, 37.0], [75, 54.0], [1, 0.6], [1, 0.7], [71, 54.0], [1, 0.7], [1, 0.5], [11, 12.0], [11, 15.3], [1, 0.6], [74, 51.0], [73, 52.0], [5, 7.5], [75, 52.0], [1, 0.6], [11, 13.1], [14, 8.0], [2, 0.7], [8, 5.0], [20, 9.0], [2, 0.6], [24, 29.8], [72, 53.0], [19, 18.1], [1, 0.6], [25, 16.0], [12, 13.0], [18, 26.2], [18, 24.4], [10, 10.0], [22, 20.8], [13, 19.2], [12, 17.8], [73, 54.0], [2, 2.8], [3, 4.6], [14, 19.2], [28, 17.8], [1, 0.7], [71, 55.0], [73, 52.0], [17, 11.2], [19, 20.0], [10, 5.0], [25, 16.0], [1, 0.6], [30, 40.0], [27, 19.9], [3, 4.0], [3, 3.4], [19, 25.7], [16, 17.0], [1, 0.5], [24, 13.0], [10, 5.0], [5, 7.5], [7, 4.5], [1, 0.5], [17, 16.3], [10, 4.0], [15, 19.0], [19, 20.0], [1, 0.6], [29, 38.7], [13, 12.7], [7, 10.1], [1, 0.7], [6, 4.0], [24, 27.4], [22, 34.0], [29, 12.6], [22, 18.6], [2, 3.8], [21, 13.6], [10, 6.0], [12, 13.0], [1, 0.6], [1, 0.5], [2, 0.5], [25, 21.0], [2, 0.5], [26, 29.6], [1, 0.7], [26, 21.8], [2, 0.6], [7, 11.5], [22, 31.8], [23, 33.2], [2, 0.6], [15, 5.5], [30, 37.0], [2, 0.7], [1, 0.7], [25, 18.5], [20, 17.0], [2, 0.5], [24, 37.0], [29, 32.9], [1, 0.5], [6, 7.0], [74, 55.0], [5, 3.5], [6, 10.0], [73, 53.0], [19, 25.7], [21, 30.4], [8, 7.4], [21, 19.9], [9, 8.2], [4, 5.8], [72, 53.0], [2, 3.2], [15, 19.0], [23, 24.0], [24, 25.0], [19, 16.2], [24, 34.6], [1, 0.6], [72, 52.0], [29, 32.9], [1, 0.7], [2, 0.6], [75, 55.0], [25, 31.0], [2, 0.7], [21, 22.0], [12, 4.6], [25, 11.0], [2, 0.5], [1, 0.5], [26, 34.8], [2, 0.5], [1, 0.6], [13, 7.5], [21, 9.4], [75, 53.0], [9, 8.2], [25, 31.0], [20, 21.0], [2, 0.5], [1, 0.6], [2, 3.6], [5, 8.5], [15, 10.0], [11, 4.3], [30, 34.0], [1, 0.6], [2, 0.6], [1, 0.5], [21, 13.6]]