

# Première NSI - Algorithmique

Les algorithmes gloutons - 5. TD

qkzk

2020/08/01

## 1. Éclairage

Il y a une longue allée dans le jardin d'Alice avec  $n$  marches d'escalier placées de manière irrégulière aux positions  $x_i$  pour  $i \in 1 \dots n$ . Elle souhaite donc installer des lanternes sur le chemin pour éviter que ses invités ne choient s'ils s'aventurent au clair de lune. Elle a repéré un modèle de lanternes de portée  $p$  qui lui plaît bien mais qui est un peu cher.

1. Proposer un algorithme glouton pour aider Alice à placer un nombre minimum de lanternes permettant d'éclairer chaque marche.
2. Votre algorithme est-il optimal ? Justifiez.
3. En combien d'étapes au maximum s'arrête votre algorithme ? Il ne reste que  $k$  lanternes en stock lorsqu'Alice fait ses courses.
4. Alice aura-t-elle assez de lanternes pour éclairer tout le chemin ?

Supposons maintenant qu'une de ses amies, médium, soit capable, sans calculer le nombre minimum de lanternes, de décider si un nombre  $k$  de lanternes est suffisant lorsqu'on lui donne la liste des positions des marches. Alice peut la consulter autant de fois qu'elle le souhaite. 5. Proposer un algorithme le plus efficace possible pour aider Alice à trouver le nombre minimum de lanternes permettant d'éclairer chaque marche.

## 2. Menu

Alice a enfin fixé sa liste d'invités et pense maintenant au menu. Malheureusement ses invités sont très difficiles, mais elle connaît l'ensemble  $s_i$  de plats parmi les  $n$  qu'elle sait préparer et qui conviennent à l'invité  $i$  pour  $i \in 1 \dots m$ . Consciente qu'elle n'aura peut être pas le temps de préparer les  $n$  plats, Alice aimerait trouver un ensemble de plats qui assure que chaque invité trouvera son bonheur sur le buffet.

1. Proposer un algorithme glouton pour aider Alice à faire son menu.
2. Votre algorithme est-il optimal ? Justifiez
3. En combien d'étapes au maximum s'arrête votre algorithme ?

## 3. Déménagement

Alice doit déménager et pour cela ranger toutes ses affaires dans des cartons. Pour faire le moins de trajets possibles dans l'escalier elle souhaite réussir à utiliser le moins de cartons possible. Cependant elle estime ne pas pouvoir porter des cartons de plus de  $C$  grammes. Alice connaît le poids  $p_i$  de chacun des  $n$  objets à placer dans les cartons, pour  $i \in 1 \dots n$  et dispose d'autant de cartons identiques qu'elle le souhaite.

1. Proposer un algorithme glouton pour aider Alice à faire ses cartons.
2. Votre algorithme est-il optimal ? Justifiez
3. En combien d'étapes au maximum s'arrête votre algorithme ?

## 4. Ranger les malles du Père Noël

Le Père Noël est en plein préparatif de la grande nuit. Le doit ranger dans les malles de son traîneau les cadeaux destinés aux enfants. Il veut optimiser son rangement de façon à utiliser le moins de malles possibles.

Pour simplifier, chaque cadeau est modélisé par son nom et sa taille.

Chaque malle est modélisée par une liste de cadeaux, et le traîneau sera modélisé par une liste de malles.

1. Exemple

```
train = {"nom": "train", "taille": 18}
nours = {"nom": "peluche", "taille": 48}
stylo = {"nom": "stylo", "taille": 1}
velo = {"nom": "velo", "taille": 25}
liste_cadeaux = [train, nours, stylo, velo]
malle1 = [train, stylo]
malle2 = [velo, {"nom": "console", "taille": 5}]
malle3 = [nours]
traîneau = [malle1, malle2, malle3]
```

- a. Combien de cadeaux y a-t-il dans `liste_cadeaux` ?
- b. Combien de cadeaux y a-t-il dans `malle2` ? c Combien de malles y a-t-il dans `traîneau` ?

2. On ajoute la contrainte suivante : chaque malle a une contenance maximale, les cadeaux qu'elle contient ne peuvent totaliser une taille supérieure à

50.

- a. Montrer que la place occupée par les cadeaux dans `malle2` est égale à 30.
- b. Quelle est la contenance disponible de `malle1` ?
- c. Écrire le code de la fonction `place_occupee` qui prend en paramètre une malle et qui renvoie la place occupée par les cadeaux de cette malle. Écrire au moins un test. On ne demande pas la documentation dans cette question.
- d. Écrire le code de la fonction `place_disponible` qui prend en paramètre une malle et qui renvoie la contenance disponible dans cette malle. Écrire au moins un test. On ne demande pas la documentation dans cette question.

3. Pour minimiser le nombre de malles nécessaires, les lutins du Père Noël décident d'utiliser l'algorithmes glouton suivant :

Données :

- une liste de cadeaux.

Résultat : un traîneau, c'est-à-dire une liste de malles.

Algorithme :

1. Initialiser `traîneau` à une liste vide.
2. Initialiser `nouvelle_malle` à une liste vide.
3. Ajouteur `nouvelle_malle` au `traîneau`.
4. Pour chaque cadeau de la liste : Si la taille du cadeau est inférieure à la contenance disponible de `nouvelle_malle` ajouter le cadeau à `nouvelle_malle`.

Sinon :

- créer une `nouvelle_malle` dans laquelle on mettra le cadeau ;
- ajouter cette `nouvelle_malle` au traîneau.

5. Renvoyer `traîneau`

Que renvoie cet algorithme avec la `liste_cadeaux` donnée à la question 1. ?

Montrer que, pour cet exemple, l'algorithme ne donne pas la solution optimale.

4. Proposez une implémentation en Python de l'algorithme des lutins. Écrire au moins un test. On ne demande pas la documentation dans cette question.