Découpe d'une corde

par David Latreyte

Un magasin achète des cordes d'escalade de longueur L et les découpe (soigneusement) en cordes plus petites pour les vendre à ses clients. On souhaite déterminer un découpage optimal pour maximiser le revenu, sachant que les prix de ventes p_i d'une corde de i mètres sont donnés.
Par exemple, supposons qu'on dispose d'une corde de $L=10$ mètres, avec les prix de ventes indiqués dans le tableau suivant :

i	0	1	2	3	4	5	6	7	8	9	10
p_i	0	1	5	8	9	10	17	18	20	24	26

1 Stratégie gloutonne

1) Rappeler en quoi consiste une stratégie gloutonne.

2) On définit la *densité* d'une corde de longueur i comme étant le rapport p_i/i, c'est-à-dire son prix au mètre. Expliquer en quoi cette grandeur pourrait être pertinente dans la mise en œuvre d'un algorithme glouton.
Remarque : une grandeur comparable a été utilisée dans le problème du sac à dos.

3) Quelle instruction mathématique permettrait de déterminer par combien de segments de longueur i une corde de longueur L peut être découpée ?

4) Quelle instruction mathématique permettrait de déterminer la longueur de corde restante après la découpe de segments de longueur i ?

5) Écrire le code de la fonction `sol_gloutonne` dont la spécification est :

```
def sol_gloutonne(L, s, p):
    """
    Détermine grâce à une stratégie gloutonne la découpe optimale
    d une corde, connaissant les longueurs des segments possibles
    et les valeurs associées.

    Paramètres
    ----------
    L : int
        Longueur de la corde
    s : list[int]
        Liste des segments possibles
    p : list[int]
        Prix associés aux segments

    Valeur retournée
```

```
    ----------------
    (somme, {segment : nombre})
        Tuple formé de la somme gagnée "optimale"
        et du dictionnaire des segments de découpe accompagnés de leur nombre.
    """
```

Utiliser les deux instructions suivantes dans le corps de la fonction (se renseigner sur l'influence du paramètre `key` dans la définition de la fonction `sorted`) :

```
donnees = [(s[i], p[i], p[i] / s[i]) for i in range(1, len(s))]
donnees = sorted(donnees, key=lambda donnee: donnee[2])
```

2 Utilisation de la programmation dynamique

6) Rappeler quel est l'objectif de la programmation dynamique.

7) Le somme maximale obtenue, lorsqu'on utilise le $i^{\text{ème}}$ segment (de longueur i donc) pour découper une corde de longueur j est donnée par la relation :

$$S[i][j] = \begin{cases} 0 & \text{si } j = 0 \\ 0 & \text{si } i = 0 \\ \max\left(S[i-1][j]; p[i] + S[i][j-i]\right) & \text{si } j \geqslant i \\ S[i-1][j] & \text{si } j < i \end{cases}$$

où p est le prix d'une découpe de longueur i. Justifier cette relation.

Remarque : une relation comparable a été donnée dans le problème du sac à dos.

8) Représenter le tableau S pour une longueur de corde $L = 4\,\text{m}$.

9) Quelle est la valeur de la somme maximale obtenue en utilisant la programmation dynamique ?

10) Comparer le résultat précédent à celui obtenu grâce à la stratégie gloutonne. Cette dernière a-t-elle conduit à un découpage optimal ?

11) Écrire le code de la fonction `sol_dynamique` dont la spécification est :

```
def sol_dynamique(L, s, p):
    """
    Détermine en utilisant la programmation dynamique la découpe optimale
    d une corde, connaissant les longueurs des segments possibles et les
    valeurs associées.

    Paramètres
    ----------
    L : int
        Longueur de la corde
    s : List[int]
```

```
        Liste des segments possibles
    p : List[int]
        Prix associés aux segments

    Valeur retournée
    ----------------
    S : List[List[int]]
        Tableau des solutions maximales des sous problèmes.
    """
```

12) Quelle est la complexité de cet algorithme ?

13) À partir du tableau de la question 8, indiquer comment retrouver le découpage conduisant à la solution.

14) Écrire la fonction `recherche_sol` dont la spécification est

```
def recherche_sol(L, s, S):
    """
    Recherche la solution et toutes les découpes correspondant à cette solution.

    Paramètres
    ----------
    L : int
        Longueur de la corde
    s : List[int]
        Liste des segments possibles
    S : List[List[int]]
        Tableau des solutions optimales des sous problèmes.

    Valeur retournée
    ----------------
    (somme, {segment : nombre})
        Tuple formé de la somme gagnée maximale
        et du dictionnaire des segments de découpe accompagnés de leur nombre.
    """
```