Informatique - DM 2

Rappel des consignes : Vous devez rendre votre copie à la date indiquée avec le code demandé et toutes les réponses aux différentes questions. Le code doit être proprement écrit, l'indentation et la syntaxe respectées.

Avant chaque partie de code, vous devez expliquer **en bon français** les algorithmes et indiquer précisément le rôle joué par chaque variable que vous nommerez de manière pertinente (suivez les recommandations qui vous ont été données en cours et TD).

Vous devez également envoyer à l'adresse `pduclosson@ac-lyon.fr` au plus tard le dernier jour des vacances un fichier nommé `DM2_nom.py` en remplaçant `nom` par votre patronyme. Ce fichier devra comporter tout le code demandé et sera présenté comme l'a été celui du DM 1.

Une entreprise achète des câbles à un fournisseur pour les couper et ensuite les vendre. La coupe peut se faire où on veut. Le prix de vente de chaque morceau obtenu dépend de sa longueur qui est toujours un nombre entier de mètres.

Un exemple de tableau de prix pourrait être :

Longueurs	1	2	3	4	5	6	7
Prix (€)	1	3	5	6	7	8	8

Pour un câble de longueur donnée, on souhaite déterminer où le couper pour maximiser le revenu R de la vente de l'ensemble des morceaux obtenus.

1. **Introduction**

   Pour une longueur de 3 mètres, il y a ainsi 3 manières de couper :
   — $3 = 3$ qui donne un revenu de 5 €
   — $3 = 1 + 2$ qui donne un revenu de $3 + 1 = 4$ €
   — $3 = 1 + 1 + 1$ qui donne un revenu de $1 + 1 + 1 = 3$ €
   Le meilleur revenu est 5 € qui est obtenu sans couper le câble.

   (a) Déterminer, en justifiant, le revenu maximal et la (ou les) manière(s) de couper correspondante(s) pour un câble de longueur 4 m.

   (b) Faire de même pour un câble de 5 mètres de long (sans justifier cette fois-ci).

   Remarque : les valeurs obtenues à la main serviront **obligatoirement** de test aux fonctions écrites part la suite.

Problème formalisé

Pour un entier naturel non nul n, on suppose donnés des nombres p_i pour $i = 1, \dots, n$.

On souhaite déterminer une décomposition de n en une somme d'entiers n_j : $n = \sum_{j=1}^{k} n_j$

telle que $R = p_{n_1} + p_{n_2} + \cdots + p_{n_k}$ soit maximal.

On utilisera dans tout le problème la liste de prix donnée par **la variable globale** :

```
prix = [0, 1, 3, 5, 6, 7, 8, 8, 10, 8, 15, 15, 16, 10, 18, 20, 24, 25, 30, 30, 40]
```

Ainsi `prix[i]` donne le prix de vente d'un câble de i mètres.

2. **Un algorithme (très) naïf**
   (a) En parcourant un câble d'un bout à l'autre, on peut couper ou non à chaque distance entière en nombre de mètres. On obtient un schéma de coupe qui peut être représenté par une liste de booléens `s` de longueur $n-1$ telle que `s[i]` indique si il y a coupe à la distance `i+1`. On peut ainsi couper un câble de longueur n de 2^{n-1} manières différentes.
   Écrire une fonction `coupe(n)` qui renvoie la liste des schémas de coupe.
   (b) Que doit renvoyer la fonction précédente pour n=1, n = 2 ? Le vérifier. Donner `coupe(4)`.
   (c) Écrire une fonction `revenu(s)` qui renvoie le revenu de la vente des morceaux correspondant à un schéma de coupe `s`. Donner `revenu([False, True, False, False])`.
   (d) En déduire une fonction `revenu_max1(n)` qui renvoie un couple donnant le revenu maximal et un schéma correspondant. Donner `revenu_max1(8)`.
   (e) Étude de la compléxité temporelle
   N.B : Pour simplifier on considérera que toutes les opérations élémentaires se font en $O(1)$.
   i.) Déterminer l'ordre [1] de la complexité de la fonction `coupe(n)` en fonction de n.
   ii.) Déterminer l'ordre de la complexité de la fonction `revenu(s)` en fonction de $m =$ `len(s)`.
   iii.) En déduire l'ordre de la complexité de la fonction `revenu_max1(n)` en fonction de n.
   (f) Analyse empirique de la complexité
   i.) Obtenir 5 listes de temps d'exécution pour n compris entre 10 et 20.
   ii.) En déduire une liste de temps moyens pour n compris entre 10 et 20.
   iii.) En déduire un tableau de valeurs donnant le quotient des temps moyens par la fonction obtenue en (e) iii.). puis le normaliser en divisant chaque élément par la moyenne de l'ensemble.
   iv.) Donner le tableau et tracer le graphe correspondant. L'expérience est-elle en accord avec la théorie ?

3. **Une propriété récursive du revenu maximal**
   Pour améliorer la complexité pitoyable de l'algorithme précédent on va avoir besoin d'établir une propriété de la fonction qui associe à une longueur n le revenu maximal possible r_n.
   On note r_n le revenu maximal possible pour la découpe d'un câble de longueur n.
   Montrer qu'en posant $r_0 = 0$, on a : $r_n = \max_{1 \leqslant i \leqslant n} (p_i + r_{n-i})$. $(*)$
   Cette propriété va nous permettre une approche par **programmation dynamique.**

4. **Un meilleur algorithme**
   (a) Écrire [2] une fonction `r_max(R)` prenant en entrée la liste `R` des revenus maximaux pour les câbles de longueur strictement inférieure à n et renvoie le revenu maximal pour un câble de longueur n calculé à l'aide de la formule $(*)$.
   (b) En déduire une fonction `revenu_max2(n)` qui renvoie le revenu maximal pour un câble de longueur n. Donner `revenu_max2(8)`.
   (c) Quelle est l'ordre de la complexité de `revenu_max2(n)` ?

5. **Avec une solution effective** **(facultatif)**
   L'algorithme précédent donne de manière efficace le revenu maximal mais ne dit pas comment couper le câble. On se propose ici de remédier à ce manque.
   (a) En modifiant la fonction `r_max(R)`, écrire une fonction `r_max_coupe(R)` qui renvoie un couple donnant le revenu maximal et la longueur du premier morceau à couper.
   (b) En déduire une fonction `revenu_max_coupe(n)` qui renvoie le revenu maximal pour un câble de longueur n ainsi qu'une liste de longueurs correspondant à une solution maximale.

1. Répondre à l'aide d'un O en **justifiant**.
2. Sans utiliser la fonction `max` de python.