Examen de
Complexité

Durée : 1 h
Tous documents autorisés
7 novembre 2017

— Les réponses aux questions doivent être dûment justifiées.
— Les programmes demandés peuvent être écrits avec le langage de programmation de votre choix.
— Le barème est donné à titre indicatif.

Rendu de monnaie

Le problème du *rendu de monnaie* consiste à rendre un montant fixé avec le minimum de pièces de monnaie pour un système de valeurs fixé (en disposant d'un nombre illimité de pièces de chaque valeur). Par exemple, si le système de valeurs est constitué de pièces de 1€, 2€ et 5€, le nombre minimal de pièces pour rendre 9€ est égal à 3 : 2 pièces de 2€ et 1 pièce de 5€.

Plus formellement, on notera :
— $v_i, \forall i \in [1, n]$ le système de valeurs considéré constitué de n différents types de pièce ;
— s le montant à rendre.

Soit $x_i, \forall i \in [1, n]$ le nombre de pièces de valeur v_i d'une solution, le problème consiste à minimiser :

$$\sum_{i=1}^{n} x_i$$

tout en respectant :

$$\sum_{i=1}^{n} x_i v_i = s$$

En notant $M(s)$ le nombre de pièces minimal pour rendre le montant s avec le système de n valeurs $v_i, \forall i \in [1, n]$, on peut résoudre ce problème grâce à la relation de récurrence suivante :

$$\forall k > 0, \quad M(k) = 1 + \min_{\forall i \in [1,n] \text{ t.q. } v_i \leq k} M(k - v_i)$$

En effet : pour rendre le montant s, il faut au moins une pièce, par exemple de valeur v_i ; en supposant que cette pièce fasse partie de la solution optimale, le nombre de pièces de celle-ci est égal à $1+M(s-v_i)$ en comptant 1 pour la pièce choisie et $M(s - v_i)$ pour ce qu'il reste à rendre. Il faut donc calculer le mininum de cette expression quelle que soit la pièce choisie parmi les n possibles.

1. [6pt] Écrire une fonction **récursive directe** qui calcule $M(s)$ en prenant en paramètres le tableau (ou la liste) des valeurs du système et le montant à rendre.
2. [4pt] Calculer les **complexités temporelle et spatiale** en pire cas de cet algorithme récursif en fonction de n et de s, en considérant que $\forall i, v_i = 1$.
3. [6pt] Écrire une fonction avec les mêmes paramètres que la fonction précédente, mais qui utilise la technique de **programmation dynamique** pour calculer $M(s)$.
4. [2pt] Calculer les **complexités temporelle et spatiale** en pire cas de l'algorithme de programmation dynamique.
5. [2pt] Exprimer la complexité temporelle de l'algorithme de programmation dynamique calculée à la question précédente en fonction de la taille k nécessaire pour encoder l'entier s en base 2. Il a été démontré que ce problème est NP-difficile ; vient-on de découvrir que finalement $P = NP$ (et gagner au passage les $1M du Clay Mathematics Institute) ?