

IFT2125: Introduction à l'algorithmique

Exercices pour les travaux pratiques

Algorithmes de programmation dynamique

1. **Problème 8.1, p.278, livre Brassard-Bratley:** Démontrez que le nombre total d'appels récurifs fait par l'algorithme suivant de calcul directe de $\binom{n}{k} = C(n, k)$ est $2\binom{n}{k} - 2$:

Algorithme $C(n, k)$ (pour $0 \leq k \leq n$)

SI $k = 0$ ou $k = n$ ALORS

RETOURNER 1

SINON

RETOURNER $C(n - 1, k - 1) + C(n - 1, k)$

2. **Problème 8.9, p.278, livre Brassard-Bratley:** Adaptez l'algorithme de programmation dynamique pour le retour de la monnaie pour qu'il fonctionne correctement même si on a un nombre limité de pièces d'une certaine valeur.
3. **Problème 8.10, p.278, livre Brassard-Bratley:** Dans le cours, pour le problème du sac à dos, nous avons fait l'exemple où la capacité du sac était $W = 11$ avec 5 objets de poids respectifs $w_1 = 1, w_2 = 2, w_3 = 5, w_4 = 6, w_5 = 7$ et de valeurs respectives $v_1 = 1, v_2 = 6, v_3 = 18, v_4 = 22, v_5 = 28$. Retravaillez cet exemple mais en numérotant les objets dans l'ordre inverse (i.e $w_1 = 7, w_2 = 5, \dots, w_5 = 1$ et $v_1 = 28, v_2 = 22, \dots, v_5 = 1$). Quels éléments de la table devraient demeurer inchangés?
4. **Problème 8.11, p.278, livre Brassard-Bratley:** Écrivez le pseudocode de l'algorithme de programmation dynamique pour le problème du sac à dos.
5. **Problème 8.12, p.278, livre Brassard-Bratley:** Pour le problème du sac à dos, quand $j < w_i$, on pose $V[i - 1, j - w_i] = -\infty$. Est-ce que la table de programmation dynamique peut contenir des entrées $-\infty$? Si oui, qu'est-ce que ces entrées indiquent? Sinon, pourquoi?
6. **Problème 8.13, p.278, livre Brassard-Bratley:** Il peut y avoir plus d'une solution optimale pour le problème du sac à dos. En utilisant la table de programmation dynamique construite pour résoudre le problème, peut-on trouver toutes les solutions optimales ou seulement une seule? Si oui, comment? Sinon, pourquoi?
7. **Problème 8.15, p.279, livre Brassard-Bratley:** Dans le problème du sac à dos considéré en classe, nous avons assumé qu'on avait n objets numérotés de 1 à n . Supposons que l'on a plutôt n types d'objets, avec une quantité très grande d'objets de chaque type. Formellement, cela revient à relaxer la contrainte que chaque x_i soit 0 ou 1 par la contrainte où x_i doit être un entier non négatif. Adaptez l'algorithme de programmation dynamique décrit au numéro 4 pour qu'il fonctionne avec ce nouveau problème.