TD 2: Programmation dynamique

December 7, 2019

Exercice 1:

La suite de Fibonacci $\mathcal{F}$ peut être définie de la façon suivante:

$$\mathcal{F}(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ \mathcal{F}(n-2) + \mathcal{F}(n-1) & \text{sinon} \end{cases} \tag{1}$$

- Justifiez l'utilisation de la programmation dynamique par rapport à un algorithme récursif pour calculer $\mathcal{F}(n)$.
- Proposez un programme dynamique pour calculer $\mathcal{F}(n)$
- Déroulez l'algorithme pour $n = 10$.

Exercice 2:

Une compagnie aérienne a décidé de changer sa politique concernant les bagages en cabine de la façon suivante:

- chaque voyageur est autorisé à un seul sac à dos
- il n'y a pas de restriction sur le volume
- un sac à dos ne doit pas dépasser 10kg de poids

Un voyageur doit alors sélectionner des d'objets à mettre dans le sac à dos (parmi n objets) qui ne dépassent pas le poids autorisé (10kg) tout en maximisant la valeur des objets sélectionnés.

Ce problème est très connu en optimisation combinatoire sous le nom "problème du sac à dos". Un problème de sac à dos $\mathcal{P}(n, W, w_1, \dots, w_n, v_1, \dots, v_n)$ est définit de la façon suivante:

- Il y a n objets $o_1, o_2, \dots o_n$ ($n \in \mathbb{N}^*$)
- Le poids total autorisé est W

- $\forall i \in [1, n]$, le poids de o_i est $w_i \in \mathbb{N}^*$
- $\forall i \in [1, n]$, la valeur de o_i est $v_i \in \mathbb{N}^*$
- $\forall i \in [1, n]$, on note par x_i la variable booléenne (binaire) de décision qui représente le fait que o_i est sélectionné dans le sac à dos. C'est à dire:
  - $x_i = 1$ si o_i est sélectionné dans le sac à dos
  - $x_i = 0$ si o_i n'est pas sélectionné dans le sac à dos
- On cherche à
  - Maximiser $\sum_{i=1}^{i=n} v_i \times x_i$
  - Tel que $\sum_{i=1}^{i=n} w_i \times x_i \leq W$

1. Soit $F(n, W)$ le coût d'une solution optimale de $\mathcal{P}(n, W, w_1, \ldots, w_n, v_1, \ldots, v_n)$. Exprimer $F(n, W)$ en fonction de $F(n-1, W), F(n-1, W - w_n)$, et v_n.
2. Justifier l'utilisation d'un programme dynamique
3. Écrire le programme dynamique associé
4. Faites l'exécution pour $\mathcal{P}(3, 10, 7, 2, 3, 5, 11, 6)$ en affichant la table dynamique (i.e., les $F(n, W)$ calculés par l'algorithme)