TD 06 – Programmation dynamique

Exercice 1. *qrcode*

1. Donner un algorithme de programmation dynamique pour résoudre le problème suivant :

   Entrée : une matrice A de taille $n \times m$ où les coefficients valent 0 ou 1.
   Sortie : la largeur maximum K d'un carré de 1 dans A, ainsi que les coordonnées (I, J) du coin en haut à gauche d'un tel carré (autrement dit pour tout i, j, $I \leq i \leq I + K - 1$, $J \leq j \leq J + K - 1$, $A[i, j] = 1$).
2. Quelle est sa complexité ?

Exercice 2. *La sieste de Maxime*

Étant donné un tableau T de n entiers relatifs, on cherche $\max\{\forall i, j \in \{1 \dots n\} : \sum_{k=i}^{j} T[k]\}$. Par exemple pour le tableau suivant :

2	18	−22	**20**	**8**	**−6**	**10**	−24	13	3

l'algorithme retournerait la somme des éléments 4 à 7 soit 32.

1. Donner un algorithme retournant la somme maximale d'éléments contigus par une approche *diviser pour régner*.
2. Donner un algorithme retournant la somme maximale d'éléments contigus par *programmation dynamique*.
3. Comparer la complexité Asymptotique au pire cas des deux approches.
4. Peut-on adapter l'algorithme de programmation dynamique en dimension 2 ? Plus formellement, étant donnée une matrice M de $n \times m$ entiers relatifs, on cherche $\max\{\forall i, j \in \{1 \dots n\}, \forall k, l \in \{1 \dots m\} : \sum_{k_1=i}^{j} \sum_{k_2=k}^{l} M[k_1][k_2]\}$.

Exercice 3. *Dans quel état j'erre ?*

La bibliothèque planifie son déménagement. Elle comprend une collection de n livres $b_1, b_2, \dots, b_n$. Le livre b_i est de largeur w_i et de hauteur h_i. Les livres doivent être rangés dans l'ordre donné (par valeur de i croissante) sur des étagères identiques de largeur L.

1. On suppose que tous les livres ont la même hauteur $h = h_i, 1 \leq i \leq n$. Montrer que l'algorithme glouton qui range les livres côte à côte tant que c'est possible minimise le nombre d'étagères utilisées.
2. Maintenant les livres ont des hauteurs différentes, mais la hauteur entre les étagères peut se régler. Le critère à minimiser est alors l'encombrement, défini comme la somme des hauteurs du plus grand livre de chaque étagère utilisée.
   (a) Donner un exemple où l'algorithme glouton précédent n'est pas optimal.
   (b) Proposer un algorithme optimal pour résoudre le problème, et donner son coût.
3. On revient au cas où tous les livres ont la même hauteur $h = h_i, 1 \leq i \leq n$. On veut désormais ranger les n livres sur k étagères de même longueur L à minimiser, où k est un paramètre du problème. Il s'agit donc de partitionner les n livres en k tranches, de telle sorte que la largeur de la plus large des k tranches soit la plus petite possible.
   (a) Proposer un algorithme pour résoudre le problème, et donner son coût en fonction de n et k.
   (b) On suppose maintenant que la la taille d'un livre est en $2^{o(kn)}$. Trouver un algorithme plus rapide que le précédent pour répondre à la même question.