TD 05 – Programmation dynamique

Exercice 1. *Étoile des neiges*

1. **Allocation de skis aux skieurs**. Spécifier un algorithme efficace pour une attribution optimale de m paires de skis de longueur $s_1, \dots, s_m$ respectivement, à n skieurs ($m \geq n$) de taille $h_1, \dots, h_n$ respectivement, *via* une fonction (injective) $f : \{1, \dots, n\} \to \{1, \dots, m\}$, f étant optimale lorsqu'elle minimise $\sum_{k=1}^{n} |s_f(k) - h_k|$.
   Indication. Soit $A[n, m]$ ce minimum. Définir $A[i, j]$ pour des valeurs plus petites $i \leq n$ et $j \leq m$ (lesquelles ?), par une équation de récurrence (i et j font référence aux i premiers skieurs et aux j premières paires de skis, respectivement).
2. **Complexité.** Analyser la complexité (en veillant à affiner l'analyse de sorte à garantir que l'algorithme soit en $\mathcal{O}(n \log n)$ si $m = n$).
3. **Grand choix de skis.** Montrer qu'on peut avoir une meilleure complexité lorsque $n^2 = o(m)$. **Indication.** Se restreindre à $\mathcal{O}(n^2)$ paires de skis.

Exercice 2. *Multi-mecs*

On considère les polygones convexes du plan. Une triangulation d'un polygone est un ensemble de cordes qui ne se coupent pas à l'intérieur du polygone et qui le divisent en triangles.

1. Montrer qu'une triangulation d'un polygone à n côtés a $(n-3)$ cordes et $(n-2)$ triangles.

Le problème est celui de la triangulation optimale de polygones. On part d'un polygone convexe $P = \langle v_0, ..., v_n \rangle$, où $v_0, \dots, v_n$ sont les sommets du polygone donnés dans l'ordre direct, et d'une fonction de pondération w définie sur les triangles formés par les côtés et les cordes de P (par exemple $w(i, j, k) = \|v_i v_j\| + \|v_j v_k\| + \|v_k v_i\|$ est le périmètre du triangle $v_i v_j v_k$). Le problème est de trouver une triangulation qui minimise la somme des poids des triangles de la triangulation.
On définit pour $1 \leq i < j \leq n$, $t[i, j]$ comme la pondération d'une triangulation optimale du polygone $\langle v_{i-1}, ..., v_j \rangle$, avec $t[i, i] = 0$ pour tout $1 \leq i \leq n$.

2. Définir t récursivement, en déduire un algorithme et sa complexité.
3. Si la fonction de poids est quelconque, combien faut-il de valeurs pour la définir sur tout triangle du polygone ? Comparez avec la complexité obtenue.
4. Si le poids d'un triangle est égal à son aire, que pensez-vous de l'algorithme que vous avez proposé ?

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 $o(2^{kn}/n)$. Trouver un algorithme plus rapide que le précédent pour répondre à la même question.