TD 05 – Programmation dynamique

Exercice 1. *Location de skis*

1. **Allocation de skis aux skieurs**. Spécifier un algorithme efficace pour une attribution optimale de m paires de skis de longueur $s_1, \ldots, s_m$ respectivement, à n skieurs ($m \geq n$) de taille $h_1, \ldots, h_n$ respectivement, *via* une fonction (injective) $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$, 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. *Triangulation de polynômes*

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.

2. Le problème est celui de la triangulation optimale de polygones. On part d'un polygone convexe $P = \langle v_0, ..., v_n \rangle$, où $v_0, \ldots, v_n$ 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 $1 \leq i \leq n$.

   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 ?