

ACT - TP Rectangle

Q1 :

La base du rectangle est sur l'axe des abscisses, il a donc exactement deux sommets avec leurs coordonnées $y = 0$.

On peut donc exprimer la surface du rectangle de surface maximale comme ceci :

$$\max_{\forall i \in \{1, \dots, n-1\}, \forall j \in \{i+1, \dots, n\}} (x_j - x_i) * \min(\{y_{i+1}, \dots, y_j\})$$

L'algorithme du fichier « Runner.java » est un algorithme utilisant 3 boucles imbriquées :

Une pour chercher une limite gauche, une pour une limite droite, une pour une limite de hauteur.

Cet algorithme a donc une complexité de $O(n^3)$.

Pour avoir un algorithme de complexité $O(n^2)$, il faut que celui-ci fonctionne avec seulement 2 boucles imbriquées. Pour cela nous pouvons enlever celle de la hauteur et se baser sur la hauteur minimum rencontrée lors des boucles.

Q2 :

L'algorithme « Divide to Conquer » est dans le fichier « Runner2.java ».

Celui-ci fonctionne de la manière suivante : il vérifie la hauteur maximum entre les 2 côtés

« divisés ». Puis il calcule l'aire correspondante, pour ensuite faire un appel récursif, et retourner l'aire maximale trouvée. Cela nous amène à dire que l'algorithme a une complexité en $O(n \log n)$, où n est la taille du tableau (taille fixée au départ).