TP2 Rectangle

Nicolas Fernandes, Alix Delbroucq

Q1)

* un rectangle de surface maximale respectant les contraintes a nécessairement deux sommets de la forme (xi , 0),(xj , 0) avec 0 ≤ i < j ≤ n – 1. C’est le cas uniquement parce que l’on souhaite que la base du rectangle soit sur l’axe des x.
* On souhaite trouver p1, p2 et h telle que pour tout (p2-p1) \* h > (p2’-p1’)\* h’. Avec p1,p2 appartenant au tableau Tx et h appartenant au tableau Ty.
* On suppose qu’une opération prend 1ms. Dans notre algorithme nous faisons dans le pire des cas 8 opérations pour n=1. On en déduit que pour n=1 t=0,008s Comme notre complexité est en n^2 ont peut que pour n= 100 t=( 100 000 \* 8 ms )^2 t= 800 000 ms t=800 s. On ne peut donc en déduire que cette complexité n’est pas compatible avec code chef.

Q2)

Cet algorithme semble plus performant que le premier. En effet, il sépare le jeu de données en deux et calcule ses parties séparément. Dans le meilleur des cas le problème est d’une complexité (n/2)² car il calcule les deux parties et fusionne les deux avec un résultat constant. Dans le pire des cas il est d’une complexité n² car il doit recalculer tout le tableau.

Q3)

Cet algorithme fonctionne uniquement s’il y a au plus un point ayant une abscisse donnée. On parcourt chaque point et on calcule l’aire entre ce point et le suivant. Lorsque le point précédent a une ordonnée inferieure au point actuel, on l’ajoute dans une pile. S’il est supérieur on le retire de la pile et on calcule l’aire grâce au point actuel et a l’ordonnée du point retiré. On reproduit jusqu'à ce que le point se situant au sommet de la pile ait une ordonnée inferieur au point actuel. La fonction retourne l’aire du plus grand rectangle.