UFR IEEA

ACT- TP

2016–2017
Master 1 d'Informatique

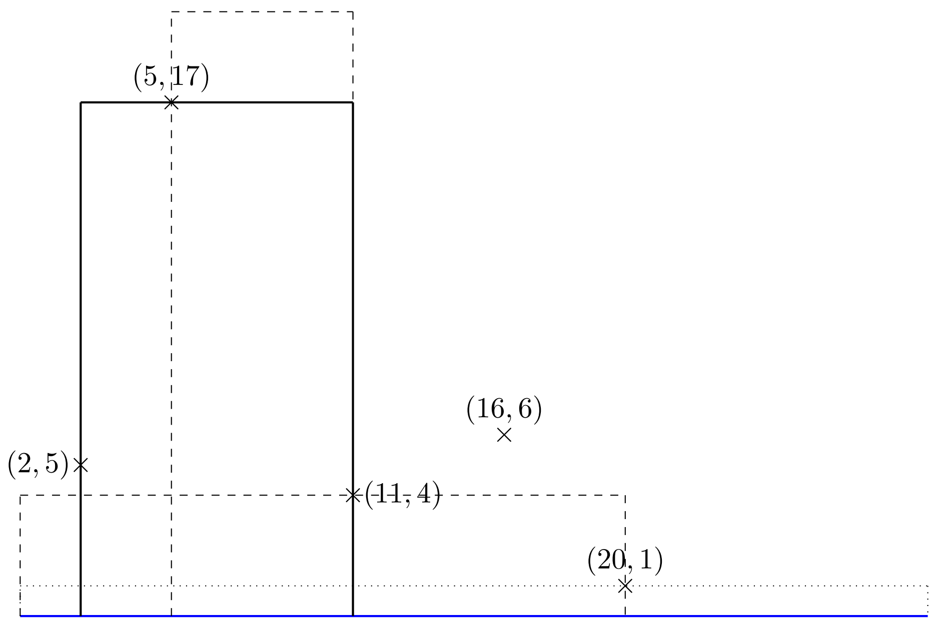
TP - Diviser pour régner
Ce sujet est fortement inspiré d'un problème du site CodeChef.

Objectif pédagogique: Ce premier TP aborde un problème d'algorithmique simple. L'objectif pédagogique est de réfléchir à différentes solutions algorithmiques tout en se posant les questions de complexité et de correction. Une façon de résoudre efficacement le problème utilise le paradigme "Diviser et Régner" .

Le problème: Soit la région rectangulaire du plan déterminée par le rectangle $(0,0)(0,h)(l,h)(l,0)$, h et l étant deux entiers positifs et soient n points à coordonnées entières à l'intérieur de cette région.

On souhaite dessiner un rectangle dont la base est sur l'axe des x, dont l'intérieur ne contienne aucun des n points et qui soit de *surface maximale.*

Par exemple si h=20, l=30, et qu'il y a 5 points (2,5), (5,17), (11,4), (16,6) (20,1), voici quelques exemples de rectangle qu'on peut dessiner: (5,0) (5,20) (11,20) (11,0) de surface 120, (0,0),(0,1),(30,1),(30,0) de surface 30, (0,0),(0,4),(20,4),(20,0) de surface 80, ... La surface maximale qu'on peut obtenir est 153 avec le rectangle (2,0) (2,17), (11,17) (11,0).



Donc l'entrée du problème est:
l, h, deux entiers strictement positifs
n, un entier s positif
les coordonnées-entières- des n points (x_i, y_i) , $0 < x_i < l, 0 < y_i < h$.

La sortie attendue est la surface maximum d'un rectangle vérifiant les contraintes.

A faire:

L'objectif du TP est de concevoir et implémenter un/des algorithmes pour ce problème, permettant si possible de traiter rapidement des données de grande taille.

Pour pouvoir tester, quelques jeux de données sont à votre disposition. Attention à la taille des entiers manipulés. Vous pouvez bien sûr aussi soumettre votre solution à CodeChef (Attention, sur Codechef l et h sont fixés, le format des entrées est donc légèrement diférent). Pour les expérimentations, vous pouvez bien sûr enrichir votre code pour mesurer le temps d'exécution.

Pour vous guider, nous vous proposons des pistes mais vous pouvez bien sûr en explorer d'autres...

Q 1. *Une première approche*

Supposons les points triés par abscisse croissante - on pourra aussi supposer (quitte à les rajouter) qu'on a un point $(0, 0)$ et un point $(0, l)$.

Une première approche peut être basée sur la remarque suivante: un rectangle de surface maximale respectant les contraintes a nécessairement deux sommets de la forme $(x_i, 0), (x_j, 0)$ avec $0 \leq i < j \leq n - 1$ (Pourquoi?).

Comment exprimer la surface du rectangle de surface maximale respectant les contraintes et dont deux sommets sont $(x_i, 0), (x_j, 0)$? Pouvez-vous en déduire un algorithme en $\Theta(n^3)$? en $\Theta(n^2)$?

Implémentez un algorithme basé sur cette approche et testez-le. Que constatez-vous? Analysez la complexité de votre algorithme. Codechef demande que l'algorithme s'exécute sur leur plateforme en moins de 1s pour n=100.000. Pensez-vous que ce soit compatible avec la complexité de votre algorithme?

Q 2. *Diviser pour régner*

En poursuivant l'approche précédente, pouvez-vous concevoir un algorithme selon le paradigme "diviser pour régner"?

Est-il plus efficace que le précédent? Testez-le sur les jeux de données proposés. Comment se comporte-t-il? Pourquoi? Quelle est ca somplexité dans le meilleur des cas? Dans le pire des cas?

Q 3. *Hauteur limitée*

Si on fixe h à une valeur relativement petite, par exemple $h = 500$, pouvez-vous améliorer votre algorithme? Expérimentez-le et analysez sa complexité.

A rendre sur Prof pour la semaine du 3 octobre: les sources de vos algorithmes, un rapport synthétique sur l'expérimentation et l'évaluation théorique de vos algorithmes.