Université de Lille 1 Sciences et Technologies
UFR IEEA

Algorithmique et ComplexiTé

2015–2016
Master d'Informatique-S1
TP 2

Diviser pour Régner : la ligne des toits

Objectif pédagogique : Ce premier TP aborde un problème d'algorithmique simple -mais non complétement trivial. 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" tout en revisitant un algorithme "classique". C'est cette solution qu'il est demandé d'implémenter.

Le problème de la ligne des toits

Un des problèmes classiques pour dessiner des images est l'élimination des lignes cachées. Ici on se place dans un cas particulier de ce problème. On se place dans le cas 2D et l'objectif est de dessiner la ligne des toits d'immeubles. Pour simplifier, on suppose que tous les immeubles correspondent (par projection) à des rectangles qui partagent tous la même base (la ville est parfaitement plate...). Un immeuble est spécifié par un triplet (g, h, d), $d > g \geq 0, h \geq 0$ qui représente donc le rectangle $(g, 0), (g, h), (d, h), (d, 0)$.

Par exemple, pour 6 immeubles donnés par (3,13,9), (1,11,5), (19,18,22), (3,6,7), (16,3,25), (12,7,16) -voir figure A, la ligne des toits sera donnée par la figure B.

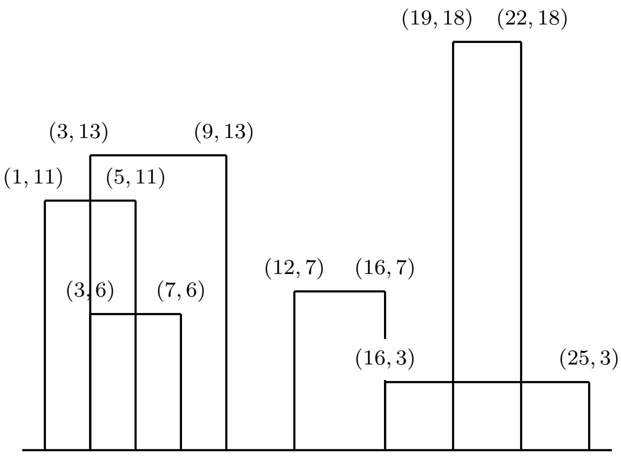


Figure A

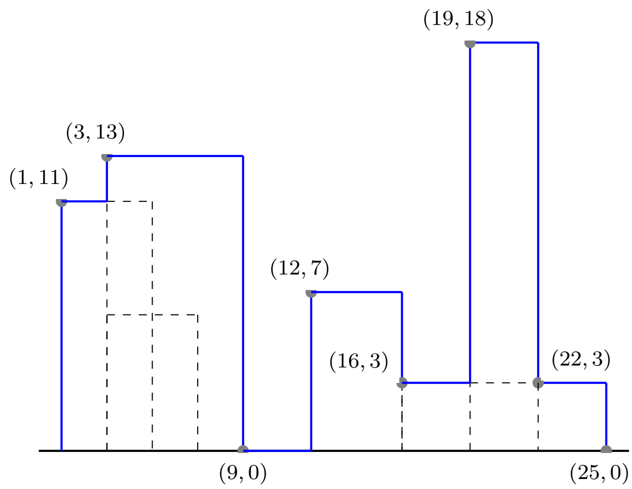


Figure B

Représentation d'une ligne de toits

Une ligne de toits est donc juste une "polyligne"[1] et sera représentée par une suite de couples (x,y). Bien sûr, cette suite est contrainte. On supposera que la ligne est triée selon les x croissants, i.e. cela correspond à une lecture de la gauche vers la droite des points d'intérêtes de la ligne.

Pour l'exemple ci-dessus, la ligne serait :

$(1,0)(1,11)(3,11)(3,13)(9,13)(9,0)(12,0)(12,7)(16,7)(16,3)(19,3)(19,18)(22,18)(22,3)(25,3)(25,0)$.

Q 1. Si une ligne de toits est une polyligne, la réciproque n'est pas vraie. Parmi les polylignes suivante,s lesquelles sont des lignes de toit (avec les hypothèses excluant toute fantaisie prises ici ... c.à.d. tous les murs sont "verticaux" et tous les toits sont "horizontaux" :-)).

- $(2,0)(2,5)(4,4)(4,7)(5,7)(5,0)$
- $(2,0)(2,5)(4,5)(4,7)(5,7)(5,0)$
- $(2,0)(2,5)(4,5)(4,7)5,7)(6,7)(5,0)$
- $(2,0)(2,5)(4,5)(4,8)(4,7)(5,7)(5,0)$

Q 2. Quelles sont les conditions pour qu'une polyligne soit une ligne de toits ?

On peut exploiter ces conditions pour choisir une représentation interne plus compacte d'une ligne de toits : par exemple $(1,0)(1,11)(3,11)(3,13)(9,13)(9,0)(12,0)(12,7)(16,7)(16,3)(19,3)(19,18)(22,18)(22,3)(25,3)(25,0)$ peut être représentée (cf figure B) par $(1,11)(3,13)(9,0)(12,7)(16,3)(19,18)(22,3)(25,0)$

1. ou ligne polygonale, ou ligne brisée

Q 3. Quelle ligne de toits serait représentée par $(1, 1)(5, 13)(9, 20)(12, 27)(16, 3)(19, 0)(22, 3)(25, 0)$?

La première représentation proposée a l'avantage d'être proche du format svg [2], qu'on utilisera pour la sortie. La seconde est plus compacte (et est peut-être plus simple à manipuler pour certains algorithmes ?). Vous pouvez choisir celle qui vous plait le mieux -ou toute autre représentation !.

Premières approches

Donc, le problème est

Entrée :

n, un entier positif : le nombre d'immeubles

une liste de n triplets d'entiers (g_i, h_i, d_i), $d_i > g_i \geq 0, h_i \geq 0$ représentant les immeubles.

Sortie : un fichier svg représentant la ligne de toits correspondante

Par exemple pour l'exemple ci-dessus, on pourrait avoir quelque chose comme :

```
<svg xmlns="http://www.w3.org/2000/svg" width="300" height="200" viewBox="-10 -150 200 150">
<polyline points="1,0 1,11 3,11 3,13 9,13 9,0 12,0 12,7 16,7 16,3 19,3 19,18 22,18 22,3 25,3 25,0"
stroke="blue" stroke-width="1" fill="none" transform="scale(5,-5)"/></svg>
```

Remarque : l'axe des y est dirigé par défaut avec les y croissants vers le bas. Le `scale(3,-3)` permet d'avoir les y croissants vers le haut d'une part et d'autre part de changer l'échelle. Pour vérifier l'exactitude de la solution , on pourrait aussi représenter les immeubles et la ligne de toits :

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="300" height="300" viewBox="-10 -150
200 150">
<rect x ="3" y ="0" width ="6" height="13" transform=" scale(5,-5) " />
<rect x ="1" y ="0" width ="4" height="11" transform=" scale(5,-5) " />
<rect x ="19" y ="0" width ="3" height="18" transform=" scale(5,-5) " />
<rect x ="3" y ="0" width ="4" height="6" transform=" scale(5,-5) " />
<rect x ="16" y ="0" width ="9" height="3" transform=" scale(5,-5) " />
<rect x ="12" y ="0" width ="4" height="7" transform=" scale(5,-5) " />
<polyline points=
"1,0 1,11 3,11 3,13 9,13 9,0 12 ,0 12,7 16,7 16,3 19,3 19,18 22,18 22,3 25,3 25,0 "
stroke="green" stroke-width="1" fill="none" transform=" scale(5,-5) " /></svg>
```

Pour les deux questions suivantes, il n'est pas demandé d'implémenter, juste de réfléchir aux deux approches, d'écrire les algorithmes en pseudo-code et d'évaluer l'ordre de grandeur de leur complexité.

Q 4. Une première approche peut être de construire une table d'entiers (ou booléens) représentant les pixels de la fenêtre de visualisation et pour chaque immeuble d'indiquer tous les "pixels de l'immeuble" à 1. Ensuite, il suffit de balayer intelligemment cette table pour calculer la ligne des toits. Quelle serait la complexité d'une telle approche ? Quel désavantage vous paraît-elle avoir ?

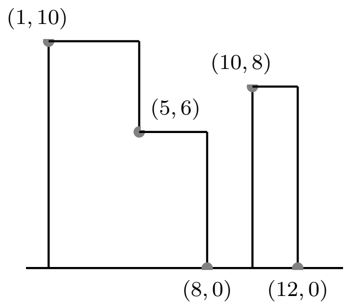
Q 5. Une deuxième approche est de construire incrémentalement la ligne en ajoutant les immeubles un à un. Proposer un algorithme pour l'insertion d'un immeuble à une ligne de toits. Quelle sera sa complexité en fonction de n, le nombre de points définissant la ligne de toits ? En déduire un algorithme pour construire la ligne des toits et analyser sa complexité.

A implémenter

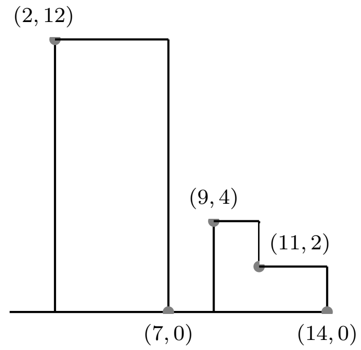
On veut résoudre maintenant le problème avec une approche "diviser pour régner", assez voisine de celle du tri fusion : on coupe la liste d'immeubles en deux listes de taille égale (à 1 près), on génère récursivement pour chacune la ligne des toits, on fusionne les deux listes de toits obtenues.

Q 6. *Fusion de lignes* On sera donc amené à fusionner deux lignes de toits en la ligne de toit correspondante. Par exemple, soit la première ligne donnée par (1,10), (5,6), (8,0), (10,8), (12,0) et la seconde par (2,12), (7,0), (9,4), (11,2), (14,0), comme dans les figures ci-dessous :

2. visualisable grâce à votre navigateur web favori

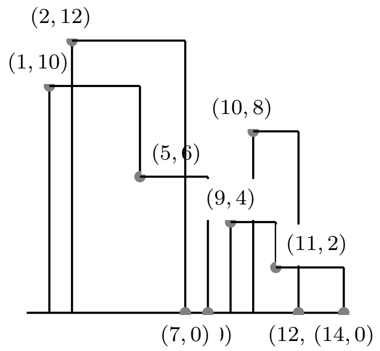


Ligne 1

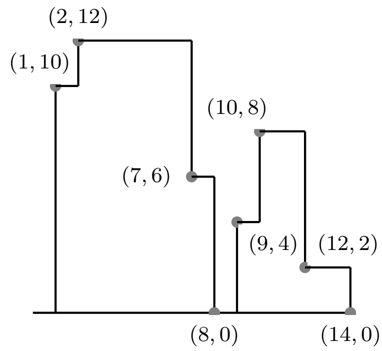


Ligne 2

La fusion des lignes des toits est (1,10), (2,12), (7,6), (8,0), (9,4), (10,8), (12,2), (14,0) :



Lignes 1 et 2



Fusion

Proposez un algorithme en $O(n)$ pour fusionner deux listes de toits, n étant le maximum des deux longueurs des lignes de toits.

Q 7. En déduire et implémenter un algorithme en $O(n \log n)$ pour la construction de la ligne des toits.