

Algorithmique Géométrique

13 mai 2005, 14h00 – 17h00 (3 heures)

Responsable : Alexandre Zvonkine. Tous documents autorisés
Barème envisagé : le barème total du problème est sur 24 points

Le monde des rectangles (suite)

Les d'exercices de cet énoncé représentent une suite naturelle des ceux proposés dans le DM, mais ils peuvent être résolus indépendamment d'eux ; toutes les notions nécessaires sont introduites de nouveau. Il est question d'ensembles de rectangles avec leurs côtés parallèles aux axes Ox et Oy (voir la figure 1).

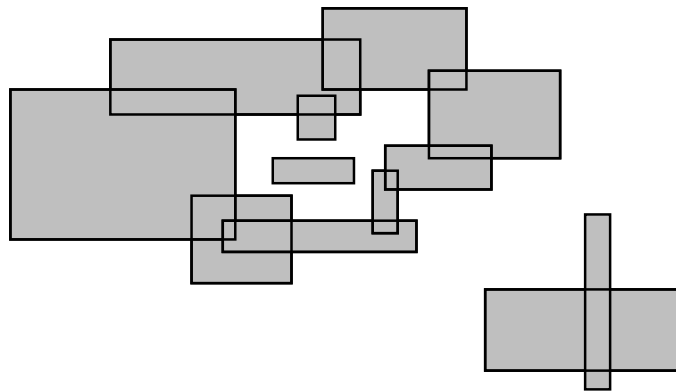


FIG. 1 – Un ensemble de rectangles sur le plan.

On va se servir d'une structure de données qui s'appelle *arbre de segments*. On suppose que l'ensemble des valeurs possibles des extrémités des segments traités est connu dès le début.

Supposons que $N = 2^k$, et les extrémités des segments traités appartiennent à une suite x_0, x_1, \dots, x_N triée en ordre croissant. Un exemple d'un arbre de segments, pour $N = 16$, est donné sur la figure 2. Un nœud de l'arbre dénoté $\boxed{i-j}$ sur le dessin, contient deux coordonnées x_i et x_j ainsi que des informations supplémentaires qui dépendent du problème traité. Le

segment $[x_5, x_{12}]$, par exemple, qui ne correspond pas à un nœud de l'arbre, est alors représenté comme union des segments $[x_5, x_6]$, $[x_6, x_8]$ et $[x_8, x_{12}]$. Il faut bien noter le fait que, une fois le segment $[x_8, x_{12}]$ utilisé, on n'utilise plus ses plus petits sous-segments. En termes généraux, un segment $[x_i, x_j]$ se représente comme union des segments qui se caractérisent par les trois propriétés suivantes :

- (a) ils font partie de $[x_i, x_j]$;
- (b) ils correspondent aux nœuds de l'arbre ;
- (c) ils sont les plus grands possibles satisfaisant (a) et (b).

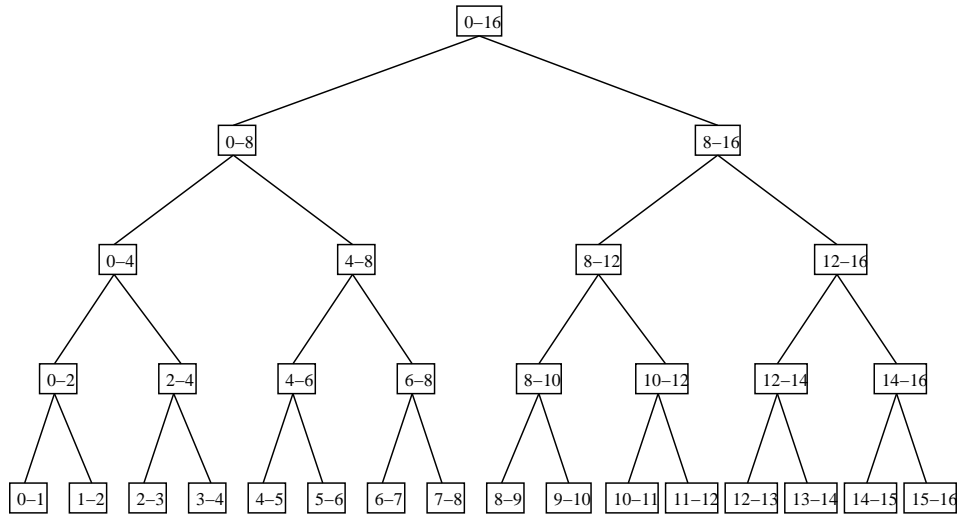


FIG. 2 – L'arbre de segments pour traiter les segments dont les extrémités appartiennent à un ensemble donné de points x_0, x_1, \dots, x_{16} . Chaque nœud, à part deux coordonnées, contient aussi des informations supplémentaires qui dépendent du problème traité.

Dans les solutions des exercices qui suivent, pour chaque opération de nature géométrique (comme, par exemple, une insertion ou une suppression d'un rectangle) il faut expliquer comment elle se traduit en opérations sur les nœuds de l'arbre de segments.

Aussi, pour tout problème traité, les données d'entrée se composent d'un ensemble de n rectangles (et de rien d'autre). Il faut donc expliquer à quel moment on construit l'arbre en question, quelle information doit-il contenir, et quelle est la complexité de sa construction.

Exercice 1. (Périmètre) Proposer un algorithme pour calculer le périmètre de l'union d'un ensemble de n rectangles. Complexité $O(n \log n)$.

Attention ! Le périmètre de l'union *n'est pas* égal à la somme des périmètres des rectangles. Les “morceaux” des segments qui se trouvent à l'intérieur de la région hachurée ne sont pas pris en compte (voir de nouveau la figure 1) : seule la frontière entre la région hachurée et la région non hachurée représente le périmètre.

Indications :

1. L'algorithme procède par balayage de gauche à droite. Événements : rencontres avec les côtés verticaux des rectangles.
2. Pour insérer un rectangle rencontré, on ajoute 1 aux compteurs dans les nœuds de l'arbre de segments représentant le côté vertical du rectangle. Pour supprimer un rectangle, on fait l'opération inverse.
3. Pour bien calculer le périmètre, il faut tenir compte des segments verticaux aux moments d'événements, et aussi des segments horizontaux entre deux événements consécutifs. Une information supplémentaire peut être envisagée pour mener à bien cette sorte de calcul ; si c'est le cas, il faut expliquer comment cette information est renouvelée aux moments d'événements.

Exercice 2. (Composantes connexes) Proposer un algorithme pour calculer le nombre des composantes connexes de l'union de n rectangles donnés (dans la figure 1, il y a trois composantes connexes).

Remarque : On peut utiliser un algorithme de recherche des composantes connexes dans un graphe. Dans ce cas-là, on n'est pas obligé de donner une présentation détaillée de cet algorithme, mais il faut dire clairement de quel algorithme il s'agit, quelle est sa complexité et, le plus important, comment le graphe en question se construit à partir de l'ensemble de rectangles.

Exercice 3. (Coins sud-ouest) Proposer un algorithme pour calculer le nombre des “coins sud-ouest” de l'union de n rectangles. Quelle est la complexité de votre algorithme ?

(Les coins de rectangles se trouvant à l'intérieure d'autres rectangles ne comptent pas. Sur la figure 3, les coins sud-ouest sont marqués par des petits ronds : il y a 3 rectangles mais 2 coins sud-ouest. Sur la figure 1, il y a 12 rectangles mais seulement 6 coins sud-ouest.)

Exercice 4. (Impasses nord-est) Un piéton peut se déplacer dans la région non hachurée vers le nord ou vers l'est, et pas dans d'autres directions. Une "impasse nord-est" est un tel point qu'un piéton se trouvant dans ce point ne peut plus bouger. Sur la figure 3, il y a une impasse nord-est : elle est marquée par une flèche. Sur la figure 1, il y a 4 impasses nord-est.

Proposer un algorithme pour calculer le nombre d'impasses nord-est pour un ensemble donné de n rectangles.

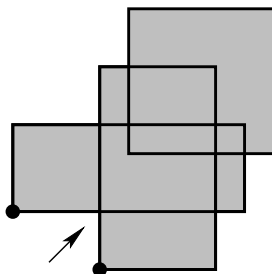


FIG. 3 – Les coins sud-ouest sont marqués par des petits ronds ; l'impasse nord-est est marqué par une flèche.