TABOULOT Clément 1A

COURET Nathanaël

**Rapport: TP n°1**

**Lecture de dallage dans un fichier:**

Cas d'erreurs identifiés:

- le fichier est vide;

- l'entête (la première ligne) ne contient pas le nombre ligne et le nombre de colonne;

- une ligne ne contient pas le bon nombre de colonne;

- le fichier ne contient pas le bon nombre de ligne;

Line test: make test\_read

**Solution 1:**

Principe:

On cherche tous les rectangles possibles à partir de chaque case du dallage. C'est-à-dire que si l'on considère une case c(x,y), on cherche tous les rectangle ayant comme case supérieure gauche c, dont la largeur varie entre 1 et width\_dallage - c\_x et dont la hauteur varie entre 1 et height\_dallage - c\_y. Pour chaque rectangle on vérifie qu'il ne contienne pas une case marquée par un '1'. S'il n'en contient pas on calcul son aire. Si cette aire est supérieure au plus grand rectangle connue, on met à jour les caractéristiques du plus grand rectangle (source\_x, source\_y, width et height).

Coût en temps:

L'algorithme contient 3 x 2 boucles imbriquées:

- n°1 : parcours de toute les cases de la dalle = changement de case de référence pour la recherche de tout les rectangles: n^2.

- n°2 : parcours de toutes les largeurs et de toutes les hauteurs possible pour une case donnée. Même si ce parcours diminue au fur et à mesure que la case de référence se rapproche du bord droit du dallage on peut considérer un coüt de n^2.

- n°3 : boucle qui vérifie si une case '1' est présente dans un rectangle donné. Cette étape est variable. En effet le parcours du rectangle ne se fera pas forcément en entier. L'arrêt du parcours dépend de la proportion de case noir présente sur le dallage. C'est-à-dire la densité de case noir présente sur le dallage. Le coût moyen de cette boucle est donc égale à 1/d(noir). Le meilleur cas (quand la dalle est toute noire) il est de 1 (constant) et dans le pire cas (quand la dalle est toute blanche) il est de n^2.

En conclusion le coût est le suivant:

Coût en mémoire:

Pour cet algorithme il n'y a aucune structure de donnée qui évolue. Il y a constamment 1 tableau de taille n\*n et les caractéristiques du plus grand rectangle. On peut donc considérer que le coût en mémoire ce cet algorithme est constant.

**Solution 2 :**

Principe :1

L’algorithme énumère tous les coins supérieurs gauches possibles pour un rectangle, i.e. toutes les cases blanches. Pour chaque coin, le dallage est parcouru ligne par ligne jusqu’à rencontrer une case noire dans la colonne du coin. La hauteur est alors définie. Pour chaque ligne rencontrée on calcule le nombre de cases blanches et s’il est inférieur à la largeur de la ligne précédente on met à jour la largeur du rectangle courant.

L’aire du rectangle courant est calculée à chaque parcours de ligne puis comparée à l’aire du rectangle maximal, si elle est supérieure le rectangle maximal est remplacé.

Coût en temps :

L’énumération des coins supérieurs gauches coûte en temps car on parcourt toute la dalle.

Pour chaque coin c du rectangle rect, le calcul de la hauteur coûte

; Soit un coût d’environ.

Et pour chaque ligne la mesure de la largeur coûte

 ; Soit un coût d’environ.

Par conséquent le coût total est de.

Coût en mémoire :

L’algorithme n’exploite pas de structure de donnée annexe dont la taille varie. Par conséquent, le coût en mémoire est de .