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:

Le principe de base de cet algorithme est le suivant. On cherche tout les rectangle possible à partir de chaque case du dallage. C'est-à-dire que si l'on considère une case c(x,y), on cherche tout les rectangle ayant comme angle en haut à gauche ce point c, dont la largeur varie entre 1 et width\_dallage - c\_x et dont la hauteur varie entre 1 et largeur\_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: n^4 <= n4/d(noir) <=n^6

Coût en mémoire:

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