Groupe TP7 BP /
Groupe TP7 YR

L2

MEMBRES DU GROUPE :

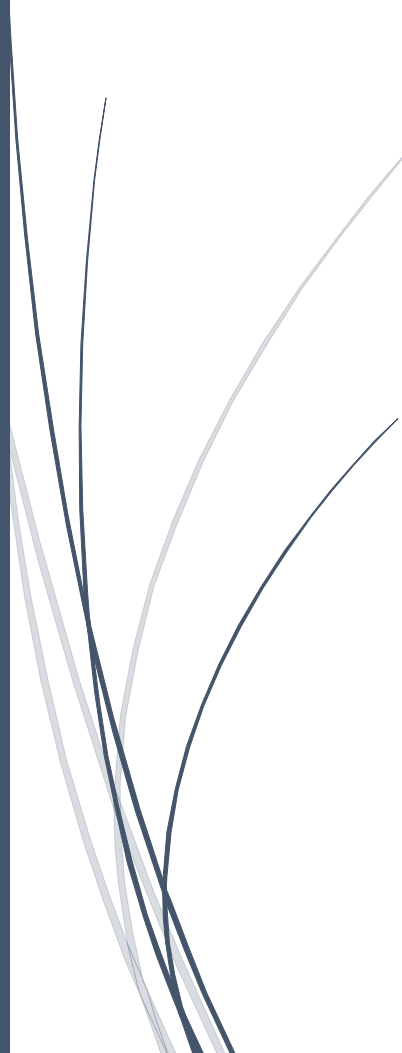
REGUEME YOHANN

PINTO Bryan

DOSSIER PROJET

Algo des Arbres

Labyrinthe



Introduction

Le projet consiste à crée un labyrinthe aléatoire à partir de la méthode union find.

Partie programmation

Le projet a été découpé en plusieurs modules :

- Le module Cellule.c / Cellule.h qui contient la structure de cellule et ses fonctions. La structure est composée de trois entiers, ligne et colonne qui représentes la ligne et la colonne de la cellule ainsi que vide (utilisé pour les cellules voisines) 1 veut dire qu'une cellule n'est pas valide (cellule hors-jeu), elle est donc vide et 0 correspond à une cellule valide. On y retrouve les fonctions Cellules_voisines qui prend en paramètre une liste de cellules et une cellule et qui modifie la liste en ajoutant les cellules voisines de la cellule. On vérifie ensuite en parcourant la liste si la cellule est valide ou non. Cellules_vois_alea qui prend en paramètres une liste de cellules et qui renvoie une cellule valide. Ainsi que d'autres fonctions explicites.

- Le module Graphique.c / Graphique.h, ce module comporte un bug d'affichage, il n'est donc pas opérationnel, il y a cependant un problème, le labyrinthe n'affiche pas la suppression étape par étape malgré la modification d'attente ou non. De ce fait, si l'option attente n'est pas en argument, le programme ne se termine pas. De plus si le labyrinthe est trop petit, il provoque une erreur de segmentation pour la fonction graphique.

- Le module Text.c / Texte.h, ce module comporte la fonction affiche_laby_texte, qui permet d'afficher un labyrinthe en version ascii sur le terminal. On a rencontré quelques difficultés lors de la conception de cette fonction car on se mélangeait dans les repères, les repères des murs horizontaux et verticaux ne sont pas les mêmes. Comme on ne compte pas les murs extérieurs, il faut ajuster la taille des listes contenant les murs.

- Le module Options.c / Options.h, qui est le module qui gère toutes les options écrites en arguments lors du lancement du programme.
(La liste de toutes les options est disponible en annexe)

- Le module Labyrinthe.c / Labyrinthe.h, Il contient la structure de labyrinthe composée d'une cellule dimensions contenant le nombre de lignes et le nombre de colonnes, d'une matrice de cellules ainsi que deux matrices d'int représentant les murs horizontaux et verticaux du labyrinthe, (1 = contient un mur, 0 = pas de murs). Il contient les structures essentielles au labyrinthe tel que init_laby, ainsi que supprime_mur et supprime_mur_opti qui est la version optimiser de supprime_mur. (Explication des changements plus loin dans le rapport)

- Le module Union.c / Union.h, il contient la structure d'union find contenant la taille du labyrinthe stockée dans une cellule, une matrice de cellule qui contient les représentants de chaque cellule ainsi qu'une matrice d'int qui contient le rang de chaque cellule. Elle comprend les fonctions init_union, qui initialiser la structure, TrouveCompresse qui trouve le père d'une cellule et compresse le chemin en même temps, de FusionRang qui effectue la fusion de deux cellules en optimisant avec le rang. On a rencontré un problème lors de sa construction, on fusionnait les deux cellules à chaque fois à la place de leur représentant. Lors de la suppression des murs, la fusion n'était donc pas prise en compte et le chemin n'était alors jamais valide ce qui donnait lieu à une boucle infinie.

Améliorations

On a effectué les améliorations suivantes :

- Toutes les options obligatoires.

- Le chemin unique, on a ajouté une condition à supprime_mur qui vérifie si deux cellules on le même représentant ou non.

- L'option acces, on a ajouté la condition nbr_suppression < n * m – 1

- Amélioration des performances on vérifie si la cellule choisit aléatoirement a déjà été supprimée ou non, si oui on en rechoisit une au hasard. Calcul du temps d'exécution avec la fonction optimisée et non optimisée.

Annexe

Pour compiler le projet, il suffit d'utiliser make, make clean supprimera les .o et make mrproper désinstallera le programme.

Les options disponibles sont :

--mode=texte, affiche le labyrinthe en ascii dans le terminal.

--taille=lignexcolonne, initialise la taille du labyrinthe.

--graine=X où x est un char * formant un nombre, permet de générer la graine du labyrinthe.

--attente=X où X est un char * formant un nombre en millisecondes, entre chaque étape de suppression, le programme attendra X temps en millisecondes. (Warning avec usleep et ansi)

--unique, qui permet d'obtenir des labyrinthe esthetiques.

--acces, qui permet de ne pas avoir de cases fermées.

--opti=X avec X = 1 ou 2. Cette option met attente à 0 et affiche le temps que le programme a mis pour générer le labyrinthe. X = 1 utilise la version optimiser de suppression, X = 2 utilise la version non optimiser de supression. Par défaut, le programme se lance avec la version optimisée.