

* Nom : RASOLOARISON

* Prénom : Tina

*TP n° : 1

*Titre : Colimacon

*temps estimé : 72 heures dont 2 heures révision des pointeurs en C

* Etudiante avec qui j'ai étudié ce devoir : Samira BELMILOUD

Informatique Embarquée

Master 2 Pro 2013-2014

Parcours Personnalisé

Rapport du TP N°1

1- Mes choix :

J'ai écrit plusieurs fonctions pour ne pas toutes les mettre dans la fonction colimacon. Ce choix est volontaire parce qu'il m'aide à trouver facilement l'endroit où une erreur apparaît(s'il y en a bien sûr). Cette modularité du programme facilite aussi sa lecture et sa compréhension.

2- Les problèmes rencontrés

L'écriture de la fonction colimacon n'était pas évidente au début, il m'a fallu passer plusieurs tests et essais avant de trouver la récursion adéquate au problème. La gestion des pointeurs en C m'a aussi pris du temps au début(2 heures) mais après que j'aie compris son fonctionnement, tout s'est bien passé.

L'algorithme que j'ai adopté pour *colimacon* est le suivant :

- J'initialise Tableau[0][0] à 1 puis je remplis le périmètre du tableau que je nomme cycle ; en incrémentant chaque valeur de 1, j'obtiens la valeur de la case suivante; l'incrémentation se fait dans le sens de l'aiguille d'une montre, je récupère la dernière valeur de ce cycle parce qu'elle me sert de repère pour le nouveau sous-tableau.
- je refais la même démarche pour le périmètre du sous-tableau à l'intérieur et ainsi de suite. Le vrai problème était de trouver le nombre de fois que je fais cette opération(nombre de cycles) en fonction du nombre de lignes M et du nombre de colonnes N. Les tests que j'ai effectués m'ont donné cette valeur que je note ici V.

soit $mini = \text{Minimum}(M, N)$

si mini est pair $V = mini / 2$

si mini est impair $V = (mini - 1)/2 + \frac{1}{2} \text{ cycle}$

3- Impact de l'algorithme vis à vis de l'accès mémoire

Plus les valeurs de rows et column sont importantes plus le programme prend de place en mémoire.

A partir de (rows =100000 et column =100000), le programme n'affiche plus rien.

Voici sont les estimations de tests que j'ai effectué :

10 * 10 : mois d'une seconde

100 * 100 : 1 seconde

1000 * 1000 : 13 secondes

10000 * 10000 : 34 minutes

100000 * 100000 : le programme ne répond plus

- **Tems estimé et temps réel de réalisation du travail**

Au début, je me suis dit que j'allais pouvoir faire ce travail en moins de 24 heures. Ce qui n'était pas vraiment le cas. J'ai dû consacrer un peu de temps au pointeur en C, j'ai passé des nuits à réfléchir sur la récursion de la fonction colimacon.c. En tout, j'ai passé en moyenne 72 heures pour ce projet sans tenir en compte les temps de discussions que j'ai eues avec certains étudiants au début du projet.

- **Lecture du travail par d'autres étudiants**

J'ai fait lire mon projet à Herinjaka Michel RABEMANANTSOA , un étudiant en Master 2 informatique et qui fait aussi informatique Embarquée. Il a dit que le programme fonctionnait bien et il a bien apprécié la modularité du programme. Par contre il n'a pas trop apprécié le fait que le programme soit long et que pour lui il serait mieux de regrouper certaines fonctions qui font presque les mêmes choses. (fonctions : incr1, incr2, incr3 et incr4).

Pour exécuter le programme sur un terminal:

gcc colimacon.c -o pgm

./pgm

Le programme ne peut pas fonctionner dans un programme multi-threads. En effet l'incrémentation des valeurs du tableau se fait de façon séquentielle (on doit toujours connaître la valeur précédente et l'incrémenter pour avoir la valeur courante). Il est quand même tout à fait possible de concevoir un algorithme qui assure la parallélisation.

Enfin ,pour tester ce programme, l'utilisateur est invité à entrer deux valeurs : ligne et colonne. Si Ces valeurs ne sont pas des entiers le programme s'arrête et une erreur apparaît.