

Compte-rendu minimal du mini-projet SDA : LCA et TH

Auteur : Fainsin Laurent

Groupe de TP : ISN-G

Building

```
$ colormake
```

Cleaning

```
$ make clean
```

Generate PDF

```
$ pandoc README.md -o README.pdf \  
-V geometry:"paperwidth=21cm, paperheight=47cm, margin=3cm"
```

Question 1.4

Implantation par listes chaînées d'une SDA.

Inconvénients

- L'utilisation de pointeurs est parfois compliquée.
- Le parcours de la liste est au minimum linéaire, et non pas en $O(1)$.
- L'utilisation des pointeurs peut mener à des situations imprévisibles voire dangereuse.

Avantages

- La taille de la liste est dynamique.

Évaluation expérimentale.

Performances LCA

```
$ build/evaluer_alea_lca 1000 1000000
```

```
Borne : 1000
```

```
Taille : 1000000
```

```
Min : 8.68000E-04
```

```
Max : 1.12400E-03
```

```
$ time build/evaluer_alea_th 1000 1000000
```

```
5,82s user 0,00s system 99% cpu 5,823 total
```

```
$ valgrind build/evaluer_alea_lca 1000 1000000
```

```
HEAP SUMMARY:
```

```
in use at exit: 0 bytes in 0 blocks
```

```
total heap usage: 3,011 allocs, 3,011 frees, 1,430,544 bytes allocated
```

```
All heap blocks were freed -- no leaks are possible
```

Performances TH

```
$ build/evaluer_alea_th 1000 1000000
```

```
Borne : 1000
```

```
Taille : 1000000
```

```
Min : 8.84000E-04
```

```
Max : 1.09400E-03
```

```
$ time build/evaluer_alea_th 1000 1000000
```

```
0,06s user 0,00s system 99% cpu 0,060 total
```

```
$ valgrind build/evaluer_alea_th 1000 1000000
```

```
HEAP SUMMARY:
```

```
in use at exit: 0 bytes in 0 blocks
```

```
total heap usage: 1,011 allocs, 1,011 frees, 710,544 bytes allocated
```

```
All heap blocks were freed -- no leaks are possible
```

Comparaison

LH est bien plus efficace temporellement et spatialement que LCA.

Qualité du générateur aléatoire

Pourvu que Taille soit assez grand, on retrouve bien un générateur aléatoire de bonne qualité puisque Min et Max sont très proches. (à noter que j'ai choisi de véritablement afficher les fréquences, puisque celles-ci doivent aussi converger vers 0 quand Taille tend vers +infini)

Principales difficultés rencontrées

La généricité des modules est parfois compliquée, il faut bien cerner le problème et cela peut demander du temps. L'implémentation en elle-même des modules n'est pas si compliqué que cela, mais le debugging (avec valgrind) pour arriver à un code sans erreur ou warning est fastidieux.

Informations complémentaires

J'ai détaillé les étapes principales du mini-projet avec git. Ainsi j'ai parfois rajouté des fonctions dans lca et th pour résoudre des problèmes liés à d'autres exercices.

Bilan personnel

Il ressemblait fortement aux quelques derniers TP/TD, donc il solidifie mes connaissances en Ada et sur les pointeurs.