

TP5 – Performances

B3125

Romain BRUNAT & Merlin NIMIER-DAVID

Test des performances

On réalise un test sur les commandes `LOAD`, `C` (création d'un cercle), `MOVE` et `SAVE`. En particulier, on crée un jeu de fichiers de test alternant une commande de création et une commande de déplacement. À l'exécution de l'application, on charge ce fichier puis on sauvegarde le résultat dans un fichier de sortie.

On utilise l'utilitaire UNIX `time` pour mesurer le temps d'exécution.

Exemple

```
C object1 0 0 1
MOVE object1 10 -10
C object2 0 0 1
MOVE object2 10 -10
C object3 0 0 1
MOVE object3 10 -10
C object4 0 0 1
MOVE object4 10 -10
C object5 0 0 1
MOVE object5 10 -10
C object6 0 0 1
MOVE object6 10 -10
C object7 0 0 1
MOVE object7 10 -10
C object8 0 0 1
MOVE object8 10 -10
C object9 0 0 1
MOVE object9 10 -10
```

Text 1: Fichier 10.in

```
LOAD 10.in
SAVE 10.out
EXIT
```

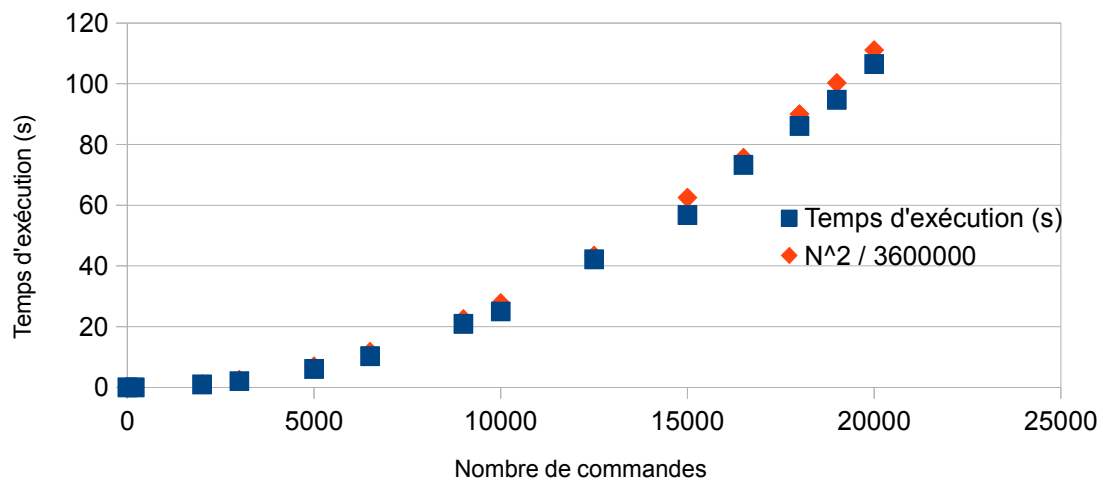
Text 2: Fichier run

```
$ time ./geometry < run
```

Text 3: Commande shell utilisée pour mesurer le temps d'exécution

Résultats

| Nombre de commandes | Temps d'exécution (s) |
|---------------------|-----------------------|
| 2 | 0.002 |
| 20 | 0.003 |
| 200 | 0.011 |
| 2000 | 0.96 |
| 3000 | 2.083 |
| 5000 | 6.1 |
| 6500 | 10.3 |
| 9000 | 20.9 |
| 10000 | 25 |
| 12500 | 42.2 |
| 15000 | 56.75 |
| 16500 | 73.3 |
| 18000 | 86.1 |
| 19000 | 94.7 |
| 20000 | 106.5 |



L'augmentation du temps de calcul est proportionnelle à n^2 , à une très grande constante multiplicative près (3600000). Asymptotiquement, le temps d'exécution des commandes testées est de $O(n^2)$. Les performances pourraient être améliorées en utilisant des algorithmes de recherche plus efficaces dans la structure de données IdSet, ou bien en choisissant une structure de donnée supportant la recherche en $O(1)$.