

# TP « Benchmark d'algorithme »

N. Delestre

L'objectif de ce TP est de développer un programme de benchmark d'algorithmes calculant la multiplication de deux `unsigned long long` en utilisant uniquement les additions, les soustractions, les multiplications et divisions par deux. Vous testerez l'algorithme classique et la multiplication égyptienne (en version itérative et récursive).

## Exemple d'exécution

```
$ bin/testMultiplier 200000000 2000000000
Multiplication classique: 200000000 * 2000000000 = 4000000000000000000
- Temps 4.023481
Multiplication égyptienne itérative: 200000000 * 2000000000 = 4000000
00000000000 - Temps 0.000001
Multiplication égyptienne récursive: 200000000 * 2000000000 = 4000000
00000000000 - Temps 0.000013
$
```

## Le projet C

Le projet C est composé des fichiers suivants :

- `include/multiplication.h` qui déclare le type `Multiplier` (pointeur de fonction qui prend en paramètre deux `unsigned long long` et qui retourne un `unsigned long long`)
- `include/multiplicationClassique.h` qui déclare la fonction *multiplicationClassique*
- `include/multiplicationEgyptienne.h` qui déclare les fonctions *multiplicationEgyptienneIterative* et *multiplicationEgyptienneRecursive*
- `src/multiplicationClassique.c` qui définit la fonction *multiplicationClassique*
- `src/multiplicationEgyptienne.c` qui définit les fonctions *multiplicationEgyptienneIterative* et *multiplicationEgyptienneRecursive*
- `src/main.c` qui outre la fonction *main* contient la fonction *benchmarker* qui prend en paramètre :
  - la fonction à benchmarker (de type `Multiplier`);
  - les deux `unsigned long long` à multiplier (issus des deux paramètres de l'exécutable);
  - le nom de l'algorithme utilisé dans la fonction.

Le script bash *compile.sh* génère :

- des `.o` dans le répertoire `src`.
- le bibliothèque `libmultiplication.a` dans le répertoire `lib`.
- l'exécutable `testMultiplier` dans le répertoire `bin`.

## Travail à réaliser

Téléchargez et décompressez l'archive `BenchmarkMultiplication.tgz` et complétez les fichiers :

- `src/main.c` de façon à benchmarker aussi les fonctions *multiplicationEgyptienneIterative* et *multiplicationEgyptienneRecursive*
- `src/multiplicationClassique.c`
- `src/multiplicationEgyptienne.c`