

Projet CMKV : Epitravex

T. Géraud

3 janvier 2023

Table des matières

1	Sujet	1
2	Sonnées et référentiel temps	2
3	Exemple	2
3.1	Fichier d'entrée	2
3.2	Fichier de sortie	2
4	Précisions	3
5	Rendu	3
5.1	Deadline	3
5.2	Mail et archive	3
5.3	Rapport	3

1 Sujet

Objectif : résoudre à l'ordinateur un problème combinatoire.

C'est basé sur le jeu Tetravex [paquet gnome-tetravex](#)

En entrée, on dispose d'un fichier texte qui donne les tuiles d'un tetravex ; ce peut-être une configuration de 2x2 tuiles à 6x6 tuiles.

En sortie, vous écrivez un fichier texte qui donne la configuration solution.

Le projet est à faire en C++ ; vous pouvez l'instrumenter d'un Makefile ou d'un CMake, mais je compilerai avec la commande : `g++ *.cc` (plus les options lisibles plus loin).

Débrouillez-vous pour que ma compil passe (donc pas de dépendance externe svp).

J'exécuterai des : `./a.out in.txt out.txt` sur un jeu de données.

Les formats d'entrée et de sortie sont décrits plus bas.

La méthode de résolution doit être un algorithme de recuit de type Metropolis-Hastings.

Le projet est à réaliser par groupe de 3 ou 4.

L'évaluation prendra en compte quatre aspects :

- facilité de compilation (les difficultés de compilation donnent des malus) ;
- justesse des solutions (pour avoir des points) ;
- rapidité d'exécution (pour avoir des bonus et pouvoir dépasser la note de 14) ;
- qualité et détail du rapport en PDF fourni dans votre archive (sur quelques points).

Les modalités dont la date de rendu sont données plus loin.

Je réponds à vos questions envoyées à thierry.geraud@epita.fr (balise [CMKV] dans le sujet du mail).

2 Données et référentiel temps

Les données d'entrée sont dans le répertoire data/input/. Les fichiers de sortie (pour que vous puissiez vérifier votre code) sont dans data/ref/.

Le chiffre en 2e caractère donne la taille de la grille.

Un temps de référence (qu'il faut que vous essayer d'approcher) est de :

— moins d'1 sec pour résoudre 70 grilles de taille 4x4

— moins de 30 sec pour 10 grilles de taille 6x6

(Pour information, j'ai un Core i7 de 8x 3,6GHz.)

3 Exemple

3.1 Fichier d'entrée

C'est un fichier texte avec une ligne par tuile, les tuiles étant listées dans l'ordre matriciel / vidéo classique (pour chaque ligne, pour chaque colonne).

Une tuile est décrite par ses 4 chiffres dans l'ordre de leur position : nord, ouest, est, sud.

Le symbole '@' en fin de ligne signifie qu'une tuile est à sa place ; sa place étant connue, cette tuile ne doit pas être "déplacée". Il peut y avoir de 1 à plusieurs tuiles bien déplacées.

Exemple de fichier de 4 lignes, correspondant donc à une configuration 2x2 :

```
0986 @
6524
5283
3805
```

représente l'entrée :

```
+-----+-----+
|  0  |  6  |
| 9 8 | 5 2 |
|  6  |  4  |
+-----+-----+
|  5  |  3  |
| 2 8 | 8 0 |
|  3  |  5  |
+-----+-----+
```

```
// ascii art rocks!
```

avec la tuile en haut à gauche restant fixée (elle reste à sa place).

Un fichier d'entrée correspondant à une grille de 6x6 aura donc 36 lignes.

3.2 Fichier de sortie

Le fichier solution attendu est de 4 lignes :

```
0986
3805
6524
5283
```

qui représente :

```

+-----+-----+
|  0  |  3  |
| 9 8 | 8 0 |
|  6  |  5  |
+-----+-----+
|  6  |  5  |
| 5 2 | 2 8 |
|  4  |  3  |
+-----+-----+

```

4 Précisions

Je compilerai avec les options `-DNDEBUG -O5 -std=c++17 -lpthread -Wl,-no-as-needed`.
Les fichiers d'entrée comporteront au moins un @ (tuile fixée).

Néanmoins, je testerai aussi avec des fichiers dont aucune tuile n'est fixée (sans @) -> ça fera des points bonus si votre programme résout le puzzle.

5 Rendu

5.1 Deadline

Le rendu s'effectue par mail à thierry.geraud@epita.fr avant le 20 janvier 2023 (à 23h51).

5.2 Mail et archive

Supposons que le nom de famille d'un membre du groupe est « <toto> ».

C'est <toto> qui m'envoie le mail à partir de son adresse en @epita.fr

Le sujet du mail commence par le tag "[CMKV]".

Attachée au mail, une archive, <toto>.tar.gz ou <toto>.zip, expande dans un répertoire qui s'appelle <toto>; ce répertoire doit contenir à sa racine :

- un fichier AUTHORS
- un fichier README.pdf (votre rapport, voir plus loin)
- des fichiers .hh et .cc directement dans ce répertoire

et c'est tout. L'archive ne doit **pas** contenir de fichiers binaires, ni de sous-répertoires.

5.3 Rapport

Dans le fichier README.pdf (en PDF donc), vous devez expliquer ce que vous avez fait : quelle est votre modélisation probabiliste, pourquoi, etc. La moitié de la note portera sur ces explications. La note augmente lorsque ces explications sont détaillées, pertinentes et compréhensibles. Le but est de me montrer que vous avez compris.