

ISAMM 2013/2014

2ème cycle ingénieur

Rapport Projet Python « Casse-Tête »

• • •

Namouchi Slim

Fathallah Ahmed

• • •

Sommaire

| | |
|----------------------------------------------------|---|
| Description de la solution..... | 2 |
| Configuration matérielle: | 2 |
| Versions de la solution : | 2 |
| Chercher tous les solutions | 2 |
| Temps d'exécution des différents fichiers : | 2 |
| Chercher Une solution..... | 3 |
| Temps d'exécution des différents fichiers : | 3 |
| Chercher tous les solutions avec la rotation | 3 |
| Temps d'exécution des différents fichiers : | 3 |
| Description des fichiers sources..... | 4 |
| Main.py..... | 4 |
| InputOutput.py..... | 4 |
| Operations.py | 4 |
| From.py | 4 |

Description de la solution

Notre solution consiste :

- Trier les formes en ordre décroissant selon leurs tailles
- Pour chaque forme tester tous les positions quel puisse l'occuper dans la matrice solution, si à partir du position $[i,j]$ on peut insérer cette forme, on insère cette forme dans cette position et on fait un appel du fonction de recherche avec l'indice de la forme suivante, si on arriver avec une indice égale au nombre des formes alors une solution est trouvé.

Configuration matérielle:

Fabricant : Lenovo

Processeur Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

RAM : 6 Go

Versions de la solution :

Chercher tous les solutions

Temps d'exécution des différents fichiers :

Entree0.txt: **0.0089 s**

Entree1.txt: **0.1563 s**

Entree2.txt **0.0284 s**

Entree3.txt **0.0788 s**

Entree9.txt **0.0952 s**

Entree10.txt **7.9400 s**

Chercher Une solution

Dans cette version, le programme s'arrête lorsqu'il trouve une solution

Temps d'exécution des différents fichiers :

| | |
|--------------|-----------------|
| Entree0.txt: | 0.0061 s |
| Entree1.txt: | 0.1178 s |
| Entree2.txt | 0.0208 s |
| Entree3.txt | 0.1171 s |
| Entree9.txt | 0.0065 s |
| Entree10.txt | 0.2372 s |

Chercher tous les solutions avec la rotation

D'abord, cette version consiste à chercher tous les rotations possibles de chaque forme et faire tous les combinaisons entre les formes originales et les nouvelles formes après rotation. Ensuite, pour chaque combinaison, chercher les solutions possibles avec la nouvelle liste des formes.

Temps d'exécution des différents fichiers :

| | |
|--------------|------------------|
| Entree0.txt: | 0.3016 s |
| Entree1.txt: | 46.9776 s |
| Entree2.txt | 3.9725 s |
| Entree3.txt | 9.1488 s |
| Entree9.txt | 4.860 s |
| Entree10.txt | s |

Description des fichiers sources

Main.py

Fichier principal, contient les appels aux différentes fonctions : lecture, recherche et écriture des résultats dans le fichier de sortie.

InputOutput.py

Fichier pour les traitements sur les entrées/sorties:

- Fonction ***lire_formes*** : La lecture du fichier d'entrée de différentes formes
- Fonction ***ecrire_formes*** : écriture des solutions dans un fichier de sortie
- Fonction ***comparer*** : comparer entre deux formes données

Operations.py

Fichier contient les traitements de base pour notre solution:

- La fonction ***insertMat*** : sert à insérer une forme dans la matrice solution à partir d'une position [i,j]
- La fonction ***testValidite*** : sert à tester si, à partir d'une position [i,j], on peut insérer une forme donnée ou non.
- La fonction ***recherche*** : c'est le cœur de notre solution, consiste à parcourir pour chaque forme toutes les positions qu'elle peut les occuper, si c'est la bonne position alors on fait un appel récursif à la fonction ***recherche*** avec l'indice de la forme suivante.

Form.py

Fichier contient la classe ***forme*** : Cette classe contient les attributs d'une forme, nombre lignes, nombre colonnes, sa taille (comptant le nombre des caractères non vides) et la matrice qui décrit la forme.