

INF4215 - Travail pratique #1

Hiver 2015

1 Description

Dans le cadre de ce premier travail pratique, vous explorerez des méthodes basées sur la recherche dans un espace d'états afin de résoudre des problèmes de Tangram. Un problème de Tangram est composé des éléments suivants :

- Une grille de dimensions $m \times n$.
- Un ensemble de pièces de formes différentes (il peut y avoir plusieurs pièces de la même forme). La forme de chaque pièce est telle que si on la place sur la grille, elle occupera entièrement un nombre exact de cases.
- Un pattern qui doit être reproduit en plaçant les pièces sur la grille.

Supposons par exemple que nous avons l'ensemble de pièces illustré à la figure 1. La figure 2 illustre deux exemples de problèmes, ainsi qu'une solution possible pour chacun.

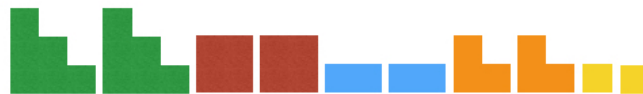


FIGURE 1 – Ensemble de pièces

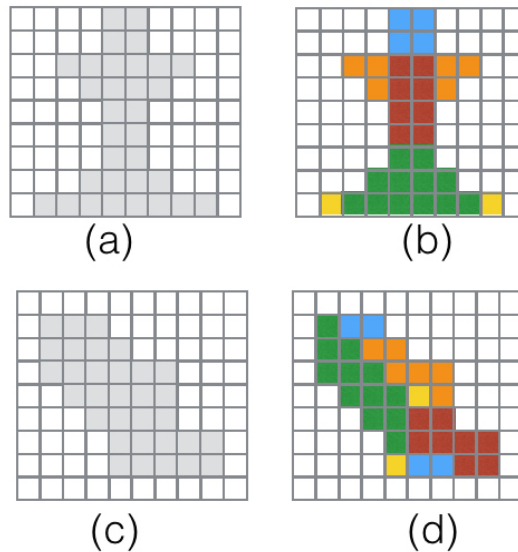


FIGURE 2 – Exemples de problèmes de Tangram

2 Travail à réaliser

Vous devrez développer deux programmes pour résoudre un problème de Tangram tel que défini ci-haut. Vous choisirez deux approches parmi les trois suivantes :

- Recherche arborescente
- Recherche locale
- Satisfaction de contraintes

Chacun de vos deux programmes doit contenir une fonction `search(Pattern, ListePieces)`. Le paramètre `Pattern` est une liste de listes, représentant la grille (chaque liste correspond à une ligne dans la grille). On utilisera la chaîne `'*'` pour représenter une case du pattern à reproduire avec les pièces, et la chaîne `' '` pour une case située en dehors du pattern. Le paramètre `ListePieces` est une liste de grilles, chacune représentant la forme d'une pièce. La fonction affiche la solution, où chaque pièce est indiquée par un identificateur distinct (un numéro de 0 à 9 ou une lettre) placé dans les cases qu'elle occupe. Voici, par exemple, comment la solution illustrée à la figure 2b serait obtenue :

d'un problème, les équipes seront divisées en cinq groupes égaux (le plus possible) et auront une note relative à leur position. Par exemple, pour 26 groupes :

| | |
|-------|-----------|
| 10/10 | 6 équipes |
| 8/10 | 5 équipes |
| 6/10 | 5 équipes |
| 4/10 | 5 équipes |
| 2/10 | 5 équipes |

4 Questions

Question 1 : Expliquez ce que fait le code suivant :

```
def fct(f, g, p, l):  
    return map(lambda x: f(g(x)),  
               [x for x in l if p(x)])
```

Question 2 : Quelles sont les points fort et les faiblesses de vos implémentations ?

5 Directives pour la remise

Le travail sera réalisé en équipe de deux personnes. Vous remettrez deux fichiers .zip (un pour chacune de vos implémentations), chacun contenant tous les fichiers Python nécessaires à l'exécution du programme. Les deux fichiers de votre remise seront nommés comme suit : *Code1_matricule1_matricule2.zip* et *Code2_matricule1_matricule2.zip*. Vous devez également remettre un fichier pdf contenant une explication de vos implémentations ainsi que les réponses aux deux questions.

Tout devra être remis avant **le 15 février à 23h55**. Tout travail en retard sera pénalisé d'une valeur de 20% pour chaque jour de retard. Le barème pour l'évaluation est le suivant :

| | |
|------------------------------|-----|
| Première méthode implémentée | 40% |
| Deuxième méthode implémentée | 40% |
| Résultat de la compétition | 10% |
| Rapport | 10% |