

# L1 Informatique

## La Grille – Projet de concrétisation

### 1 Problème

Le problème consiste à placer des pions de diverses couleurs sur un plateau de jeu (que l'on appellera *grille*), de manière à maximiser son score.

On dispose au départ des données suivantes :

1. Une grille, définie par une matrice carrée de valeurs positives et négatives.  
La *taille* de la grille ( $t$ ) est égale à son nombre de lignes (et donc de colonnes).  
Voici un exemple de grille de taille  $t = 4$  :

	1	2	3	4
A	-6	9	-2	5
B	8	2	-7	-6
C	-8	2	-4	-3
D	-3	3	-4	7

2. Une valeur de pénalité ( $p$ ). Par exemple :  $p = 10$ .

Sur chacune des  $t^2$  cases de la grille, il faut placer un pion. Le *score total* correspond à la somme du score de tous les pions, moins les points de pénalité. Chaque pénalité retranche  $p$  points au score total.

Il existe 6 couleurs de pion ; les règles de placement et la manière de marquer des points ou d'engendrer des pénalités sont propres à chaque couleur :

1. Pions **jaunes** : Le score d'un pion jaune correspond à la valeur de sa case. Tout pion jaune isolé, c'est-à-dire n'ayant aucun pion jaune parmi les pions adjacents (y compris en diagonale), entraîne une pénalité.
2. Pions **verts** : Le score d'un pion résulte de la somme des valeurs de sa case et des cases adjacentes orthogonalement (donc pas en diagonale). Chaque paire de pions verts adjacents (y compris en diagonale) entraîne une pénalité.
3. Pions **noirs** : Le score d'un pion noir est la valeur de sa case, moins 1. Si au plus  $n$  pions noirs sont utilisés, le score de chaque pion noir est doublé.
4. Pions **bleus** : Le score d'un pion bleu est 0. Soit  $d$  la différence entre le nombre de pions bleus placés sur des cases de valeur strictement négative et le nombre de pions bleus placés sur des cases de valeur strictement positive. Si  $d > 0$ , alors cela entraîne  $d$  pénalités.
5. Pions **orange** : Le score d'un pion orange est 0. Chaque paire de pions orange situés sur la même ligne, même colonne ou même diagonale entraîne une pénalité.
6. Pion **rouge**. Il n'y a qu'un seul pion rouge à disposition, et il est requis de l'utiliser. Le nombre de points correspond à l'**opposé** de la valeur inscrite sur la case.

À l'exception de l'unique pion rouge, le nombre de pions de chaque autre couleur n'est pas limité.

Ci-dessous à gauche figure un exemple de placement de jetons (pas optimal en termes de score) sur une grille de taille  $t = 4$  :

	1	2	3	4
A	V	J	O	N
B	J	V	B	O
C	R	J	O	J
D	J	N	O	B

	1	2	3	4
A	-6	9	-2	5
B	8	2	-7	-6
C	-8	2	-4	-3
D	-3	3	-4	7

Le score obtenu par ce placement de jetons sur la grille exemple précédente (rappelée ci-dessus à droite) se calcule comme suit :

- Score des pions jaunes :  $S_J = 9 + 8 + 2 - 3 - 3 = 13$ .
- Pénalités sur les pions jaunes :  $P_J = 1$ , car il y a un pion isolé (C4).
- Score des pions verts : le pion en A1 a un score de  $-6 + 9 + 8 = 11$ , le pion en B2 a un score de  $2 + 9 - 7 + 2 + 8 = 14$ , soit un score pour les pions verts de  $S_V = 11 + 14 = 25$ .
- Pénalités sur les pions verts :  $P_V = 1$ , car les pions en A1 et B2 sont adjacents.
- Score des pions noirs :  $S_N = 2 \times ((5 - 1) + (3 - 1)) = 12$ ; le score de base est doublé car il n'y a pas plus de  $t = 4$  pions noirs.
- Pénalités sur les pions bleus :  $P_B = 0$ , car il n'y a pas plus de pions bleus sur des cases négatives (B3) que sur des cases positives (D4).
- Pénalités sur les pions orange :  $P_O = 5$ , puisque sur les  $\binom{4}{2} = 6$  paires de jetons orange, 3 sont placées sur une même colonne (A3-C3, A3-D3, C3-D3) et 2 sur une même diagonale (A3-B4, B4-C3).
- Score du pion rouge :  $S_R = 8$ .
- Pour rappel, les pions bleus et les pions orange ont toujours un score nul, et il n'y a jamais de pénalité sur les pions noirs ni sur le pion rouge.
- Le score total de la solution est :  $S_J + S_V + S_N + S_R - p \times (P_J + P_V + P_B + P_O)$   
 $= (13 + 25 + 12 + 8) - 10 \times (1 + 1 + 0 + 5) = 62 - 70 = -12$ .

On appelle *solution* une configuration valide de placement des jetons sur la grille (une couleur par case, et exactement une seule fois la couleur rouge), et *solution optimale* la meilleure solution possible en termes de score. Il peut y avoir plusieurs solutions optimales, si elles sont de score identique.

Deux solutions optimales, de score 76, sont représentées ci-dessous. Outre le pion rouge requis (score 8), elle n'utilise aucun pion jaune, 3 pions verts (score 18), 4 pions noirs (score 50), 6 pions bleus et 2 pions orange. Elles diffèrent uniquement par la permutation d'un pion orange et d'un pion bleu. Ces configurations n'entraînent aucune pénalité.

	1	2	3	4
A	V	N	V	N
B	N	B	B	O
C	R	B	B	B
D	O	B	V	N

	1	2	3	4
A	V	N	V	N
B	N	B	B	B
C	R	B	O	B
D	O	B	V	N

## Syntaxe des fichiers d'entrée (instances)

Une *instance* de problème à résoudre est donnée par un fichier texte comportant, sur la première ligne, la taille  $t$  de la grille (c'est-à-dire son nombre de lignes, égal à son nombre de colonnes) et la valeur  $p$  d'une pénalité (nombre de points toujours positif). Les autres lignes comportent les valeurs des cases. Par exemple, pour une grille de taille 4, il s'agira de 4 lignes de 4 valeurs chacune.

Le fichier représenté ci-dessous correspond à la grille de taille  $t = 4$  de l'exemple précédent, avec une valeur de pénalité fixée à  $p = 10$  points.

4	10		
-6	9	-2	5
8	2	-7	-6
-8	2	-4	-3
-3	3	-4	7

## Syntaxe des fichiers de sortie (solutions)

Un fichier de solution pour une grille de taille  $t$  doit comporter, sur les  $t$  premières lignes, la suite de  $t$  caractères correspondant à l'initiale de chaque couleur de pion, dans l'ordre des colonnes. On fera figurer sur la  $(t + 1)$ -ième et dernière ligne le score de la solution.

Le fichier suivant correspond à la première solution optimale de l'exemple.

VNVN
NBBO
RBBB
OBVN
76

## 2 Projet

Il est demandé d'écrire un programme C++ permettant de résoudre le problème de *La Grille*, c'est-à-dire lire un fichier d'entrée représentant une instance de problème, calculer une solution, et l'écrire (ainsi que son score) dans un fichier de sortie. L'objectif est de concevoir un algorithme permettant de trouver une solution de plus grand score total à une instance quelconque en moins d'une minute de calcul. La solution trouvée ne devra pas obligatoirement être une solution optimale.

Attention à la complexité de l'algorithme, car la taille des grilles pourra être de plus en plus importante. Il existe, pour une grille de taille  $t$ ,  $5^{t^2-1} \times t^2$  placements possibles de jetons. Même pour une petite grille de taille 4 comme l'exemple précédent, cela représente plus de 488 milliards de solutions ; une simple énumération est donc inenvisageable.

Le site web associé au projet devra comporter en particulier les éléments suivants :

- la description du problème ;
- le ou les algorithmes développés pour sa résolution (explication de la méthode) ;
- une explication précise permettant à un utilisateur de compiler et d'exécuter votre code source ;
- les solutions et scores obtenus des instances de problèmes fournies (une fois la compétition arrêtée).

Durant le projet, des nouvelles instances de problèmes, avec des grilles de diverses tailles, seront mises à disposition sur Moodle.

### 3 Compétition

La compétition entre les différentes équipes sera jugée sur 2 épreuves<sup>1</sup> :

1. **Résultats sur les instances connues.** Avant la dernière semaine, chaque équipe devra rendre une archive comportant la meilleure solution trouvée pour chacune des 15 instances fournies au cours du projet et indiquées *pour compétition*<sup>2</sup>. Pour cette compétition, il ne sera pas vérifié que l'algorithme a trouvé cette solution en moins d'une minute, tout est donc permis (sauf voler les résultats des autres équipes...).  
→ Pour chaque instance compétitive mise à disposition sur Moodle au cours du projet, l'équipe (ou les équipes) obtenant le meilleur score marque 1 point. L'équipe (ou les équipes) obtenant le plus de meilleurs scores sur cette épreuve marquent 2 points supplémentaires ; l'équipe (ou les équipes) obtenant le *meilleur score moyen* (moyenne des 15 scores) marque 3 points supplémentaires.
2. **Résultats sur des instances mystère.** La dernière semaine, chaque équipe devra rendre un code source C++. Les algorithmes seront mis en compétition pour résoudre en moins d'une minute (par instance) 10 instances de problème qui n'ont pas été communiquées précédemment. Chaque programme ne sera exécuté qu'une seule fois par instance.  
→ Pour chaque instance, l'équipe (ou les équipes) obtenant le meilleur score marque 2 points (1 point pour la deuxième place s'il n'y a pas d'égalité pour la première). L'équipe (ou les équipes) obtenant le plus de meilleurs scores sur cette épreuve marque 4 points supplémentaires ; l'équipe (ou les équipes) obtenant le *meilleur score moyen* (moyenne des 10 scores) marque 6 points supplémentaires<sup>3</sup>.

Le classement des équipes sera établi à partir des points (score maximum : 50). Le cas échéant, les équipes à égalité seront départagées sur une instance mystère supplémentaire.

---

1. Rappel : les points obtenus à la compétition ne sont pas les points de la note finale.

2. D'autres instances indiquées *hors compétition* seront également fournies, pour lesquelles des solutions et des scores pourront être communiqués librement entre les équipes.

3. Pour marquer ces 6 points, il faut en particulier que le programme termine en moins d'une minute et calcule une solution pour chacune des 10 instances mystère.