

► MP – Option Informatique

4ème TP Caml
sl@jachiet.com
http://jachiet.com/tps
Lundi 14 janvier 2012

Taquin

Le taquin est un jeu solitaire en forme de damier créé vers 1870 aux États-Unis. Sa théorie mathématique a été publiée par l'American Journal of mathematics pure and applied en 1879. En 1891, son invention fut revendiquée par Sam Loyd, au moment où le jeu connaissait un engouement considérable, tant aux États-Unis qu'en Europe. Il est composé de 15 petits carreaux numérotés de 1 à 15 qui glissent dans un cadre prévu pour 16. Il consiste à remettre dans l'ordre les 15 carreaux à partir d'une configuration initiale quelconque. Source : Wikipedia

Dans ce TP, je vous laisse une grande liberté quant aux choix d'implémentations. Attention à ne pas vous perdre au fil du TP dans vos choix (x ou y en première coordonnée par exemple).

1 Représentation du taquin

Pour représenter un taquin on utilisera la structure suivante (x et y dénotent la position du trou) :

```
type taquin = { grille : int vect vect ; mutable x : int ; mutable y : int }
```

Par ailleurs, on note par `taille` la taille du carré. En effet, le taquin est souvent en 4×4 , mais pour des raisons de performances, on utilisera plutôt du 3×3 .

► **Question 1** Écrire une fonction qui produit un taquin dans sa position initiale (comme ci-dessous).

1	2	3
4	5	6
7	8	0

► **Question 2** Écrire une fonction qui copie une structure taquin. Attention, toute modification sur la copie ne doit pas avoir d'effet sur l'original (et vice et versa). Cette question n'est **pas** triviale!

► **Question 3** Écrire une fonction qui affiche la grille du taquin.

2 Déplacements

Il y a, au plus, quatre déplacements possibles : vers le haut, la droite, la gauche, le bas. On représente ça par la différence en coordonnées x et y du trou (le 0 dans la grille). Aussi le tableau des mouvements possibles est :

```
let déplacements = [(1,0);(0,1);(-1,0);(0,-1)];;
```

► **Question 4** Écrire une fonction qui détermine si un mouvement donné est possible.

► **Question 5** Écrire une fonction qui déplace effectivement le trou dans une structure taquin.

3 Résolution du taquin

On cherche ici à, étant donné une grille de taquin, trouver une séquence de déplacements qui le remet dans sa position initiale. Pour ne pas boucler dans la recherche d'une telle solution, nous commencerons par trouver une fonction injective qui à une grille de taquin associe un entier.

L'idée, c'est que si cette fonction envoie toutes les grilles de taquin vers des entiers inférieurs à n , alors un tableau de booléens de taille n permettra de savoir si on a déjà étudié une grille ou non.

► **Question 6** Écrire une fonction `encode` qui prend une grille de taquin et renvoie un entier. Cette fonction doit être injective et vous devez borner au mieux le plus grand entier que cette fonction peut retourner.

► **Question 7** Écrire une fonction qui recherche récursivement la solution d'un taquin. La forme générale doit ressembler à :

Créer un tableau global de booléens taille 400 000

Recherche grille

Si la grille a déjà été vue

Arrêter d'explorer cette voie

Si la grille est bien rangée

Lever une exception "trouvée"

Marquer la grille comme vue

Pour chacun des déplacements possibles

Faire le déplacement

Rechercher récursivement

Annuler le déplacement

► **Question 8** Afficher une séquence de déplacements qui résout le taquin.

4 De meilleures façons de résoudre le taquin

L'algorithme que vous venez d'utiliser s'inspire de la recherche de chemins dans un graphe. Les nœuds sont les grilles et les arêtes sont les déplacements. Pour rechercher un chemin dans un graphe, voici quelques algorithmes :

- **recherche en profondeur d'abord** (ce que vous venez de faire)
- **recherche en largeur d'abord** À l'étape $n + 1$ on

calcule les voisins des nœuds qu'on a trouvés à l'étape n . Tous les chemins sont optimaux.

- **Dijkstra** C'est une généralisation de la recherche en largeur d'abord pour les graphes où les arêtes ont des poids variables (mais toujours positifs).
- **A*** C'est une modification de dijkstra à l'aide d'une fonction potentiel. L'idée c'est de changer le poids des arêtes de sorte qu'on ne visite que peu de nœuds (on ne visite que les nœuds à distance moindre que notre objectif).

► **Question 9** *Implémenter ce qu'il vous plaît.*

► **MP – Option Informatique**
4ème TP Caml
sl@jachiet.com
<http://jachiet.com/tps>
Lundi 14 janvier 2012

Taquin

Un corrigé