

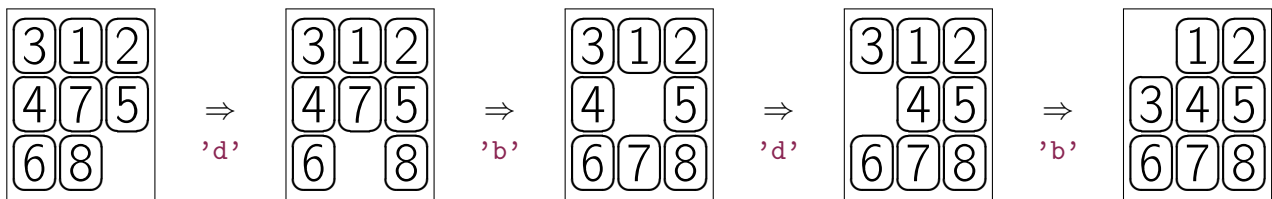
TECHNIQUES ALGORITHMIQUES ET PROGRAMMATION

TP noté – 2h10

Jeu de Taquin

On considère le Taquin d'ordre n , un jeu à un joueur composé de $n^2 - 1$ carreaux numérotés de 1 à $n^2 - 1$ qui glissent dans un cadre $n \times n$, avec une case vide donc. Il consiste à remettre dans l'ordre tous les carreaux à partir d'une configuration initiale quelconque. Un carreau ne peut se déplacer que s'il est voisin de la case vide (4-voisinage). Une fois effectué, le déplacement crée une nouvelle configuration avec une autre case vide.

Ci-dessous, un exemple de configuration de Taquin d'ordre $n = 3$ (figure de gauche), où quatre déplacements permettent de gagner, c'est-à-dire d'atteindre la configuration gagnante (figure de droite).



On représente une configuration par un tableau 2D où $G[i][j]$ est un entier unique de $[0, n^2 - 1]$, la case vide étant codée par 0. La case $G[i][j]$ correspond au carreau situé sur la i -ème ligne et j -ème colonne, $G[0][0]$ étant située en haut à gauche. C'est aussi la case vide de la configuration gagnante.

Suivant la configuration de départ, il n'est pas toujours possible d'atteindre la configuration gagnante. C'est par exemple le cas si comme configuration de départ, on choisit la configuration gagnante dans laquelle on a échangé les carreaux numérotés 1 et 2.

Chaque déplacement est déterminé par sa direction vers la case vide. Plus précisément :

- 'd' : on déplace un carreau vers la droite (\rightarrow : incrément de j),
- 'g' : on déplace un carreau vers la gauche (\leftarrow : décrément de j),
- 'b' : on déplace un carreau vers le bas (\downarrow : incrément de i),
- 'h' : on déplace un carreau vers le haut (\uparrow : décrément de i).

Remarquez qu'un déplacement modifie deux valeurs du tableau G : celle de la case qu'on déplace et celle de la case contenant la valeur 0, qui représente la case vide.

Une suite de mouvements est codée par une chaîne, comme "dbdb" correspondant aux quatre déplacements de l'exemple. Certains déplacements sont invalides, comme 'd' ou 'b' pour la configuration gagnante, car sa case vide n'a pas de carreau adjacent pouvant se déplacer vers la droite ou le bas.

L'objectif du TP est de programmer les fonctions utilisées pour la recherche de la suite de déplacements la plus courte permettant de gagner, lorsque c'est possible. On va faire cette recherche à l'aide de l'algorithme A^* vu en cours et TD. Pour cela lisez attentivement les consignes en commentaires dans le fichier `taquin.c`, éditez ce fichier, testez vos fonctions, puis soumettez uniquement ce fichier complété avant la fin de la séance sous la page Moodle du cours.

FIN.