

## Série 1

### Recherche aveugle et heuristique

#### Exercice 1 (recherche aveugle)

Coder la recherche aveugle en largeur (cf. diapositive 27) pour un puzzle-8 et un puzzle-15. A noter que la recherche en largeur est associée à une file d'attente pour la visite des états successifs. **Dans le but de pouvoir réaliser plusieurs tests, l'état initial et l'état final devront être introduits au clavier et donner aussi la possibilité de définir ces états aléatoirement.** Afficher à la fin de la recherche le chemin qui aboutit à la solution et le nombre d'états explorés.

#### Déroulement

- Pour le puzzle-8, dans une première étape on ne tiendra pas compte des états déjà visités. Dans ce cas, est-ce qu'on arrive toujours à la solution optimale, si elle existe ?
  - Même question en considérant les états déjà visités. Ils pourront être sauvegardés dans un arbre binaire ou dans une table de hachage. Est-ce que la recherche est plus rapide ?
  - Pour le puzzle-15 les états déjà visités ne seront pas considérés. Si la file d'attente des éléments à visiter dépasse un certain seuil (p. ex : cent millions d'éléments) on arrêtera l'algorithme et on affichera l'état représenté par le premier élément de la file d'attente. Avec un peu de chance on aura trouvé une solution.
- 
- **Etats** : numéros des positions des plaquettes
  - **Opérateurs** : déplacer la case vide à gauche (L), à droite (R), en haut (U), en bas (D)
  - **Test** : état courant = état final
  - **fonction-coût** : chaque déplacement de la case vide vaut 1, coût total = nombre total de déplacements de la case vide

#### Exercice 2 (recherche heuristique)

Coder la recherche heuristique (cf. diapositive 40) pour un puzzle-8 et un puzzle-15 avec dans une première version la fonction heuristique « nombre de plaquettes mal placées » et dans une deuxième version avec la fonction heuristique « somme des distances de chaque plaquette numérotée à sa position finale » (cf. diapositive 42). Ici, le nouvel état à visiter sera toujours le plus prometteur, c'est à dire celui qui est associé à la valeur la plus faible des fonctions heuristiques. Dans ce type de recherche il sera impératif de tenir compte des états déjà visités. De la même manière que dans l'exercice précédent, pour un puzzle-15 on arrêtera l'algorithme si le nombre d'éléments dans la file d'attente ou dans la structure des états déjà visités dépasse un certain seuil.

**Comparer les solutions données par les deux types de recherche en partant du même état initial (pour plusieurs états initiaux).**

**Ce travail sera noté. Il devra être rendu mardi 1<sup>er</sup> novembre 2016.**