

Recherche heuristique, représentation et résolution de connaissances

1. **Modélisation états - sur papier** *Le problème des 8 dames* : placer 8 dames d'un jeu d'échecs sur un échiquier 8x8 sans qu'elles se menacent mutuellement (donc deux dames ne doivent jamais partager la même rangée, colonne, ou diagonale).
Modélisez le problème de 8 dames en détaillant l'espace d'états, les actions, le modèle de transition, les états but (plusieurs solutions sont possibles). Détaillez les premiers deux niveaux de l'arbre de recherche après la racine à partir d'un état initial différent de celui dans les slides. Analysez l'arbre de recherche correspondant en calculant le nombre d'états, le facteur de branchage, la profondeur, et le nombre de feuilles.
2. **Recherche non informée - sur machine** Implementation et comparaison de DFS et de BFS pour le jeu du taquin : voir notebook sur l'espace du cours.
3. **Recherche heuristique - sur papier** Rappellez-vous de la modélisation en espace d'états du jeu de taquin. Considérez les deux heuristiques présentées en cours (slide 37 cours recherche heuristique) :
 - h1 : nombre de tuiles mal placées
 - h2 : distance de Manhattan avec l'état but (pour chaque tuile, compter combien de cases elle doit croiser pour arriver à sa case dans l'état but)
 Déroulez A* sur au moins 3/4 niveaux de l'arbre avec h1 et h2 à partir de l'état initial ci-dessous, et comparez visuellement avec le fonctionnement de BFS ou DFS.

1	2	3
4	5	
6	7	8

4. **Recherche heuristique - sur machine** Implementation et comparaison de A* avec deux heuristiques : voir notebook sur l'espace du cours.
5. **SAT et CSP - sur papier** Modélisez le problème de résoudre une grille de Sudoku avec des formules proportionnelles (SAT). Analysez la modélisation en terme de nombre de variables et nombre de formules.

1	2		
		2	1
2	4		
		4	2

6. **SAT et CSP - sur machine** Utilisez les librairies PySAT <https://pysathq.github.io/> et la librairie Python constraints <https://pypi.org/project/python-constraint/> pour implanter le Sudoku comme un problème de SAT et CSP : voir notebook.
7. **Extra.** Modélisez le problème de résoudre une grille de Sudoku en détaillant l'espace d'états, les actions, le modèle de transition, les états but (plusieurs solutions sont possibles). Détaillez les premiers deux niveaux de l'arbre de recherche après la racine à partir de l'état initial ci-dessous. Analysez l'arbre de recherche correspondant en calculant le nombre d'états, le facteur de branchage, la profondeur, et le nombre de feuilles.
8. **Extra.** Modélisez la recherche d'un chemin Hamiltonien dans le graphe suivant en détaillant l'espace d'états, les actions, le modèle transition, l'état but. Analysez l'arbre de recherche correspondant. Développez des heuristiques admissibles. Modélisez le problème comme un SAT et CSP. Implementez BFS, DSF, A*, SAT, CSP et comparez les temps d'exécution sur un graphe de taille suffisamment grande.