

Fiche de Suivi de Projet (SAE)

Exploration Algorithmique : Le Tour du Cavalier

BUT Informatique - Semestre 2

Phase 1 : Compréhension et Modélisation (Papier/Crayon)

Avant de coder, visualisez le problème pour comprendre la théorie des graphes sous-jacente.

- ☐ **Modéliser l'échiquier en graphe** : Comprendre que chaque case est un sommet et chaque mouvement légal (L) est une arête.
- ☐ **Définir la structure de données** : Choisir la représentation en mémoire (ex : un tableau 2D `int echiquier[8][8]` où 0 = vide et 1..64 = ordre de passage).
- ☐ **Identifier les mouvements** : Lister les 8 déplacements relatifs possibles ($x \pm 1, y \pm 2$) et ($x \pm 2, y \pm 1$) pour préparer les tableaux de direction.

Phase 2 : Implémentation de l'Algorithme (Cœur du sujet)

Objectif : Coder l'algorithme DFS (Depth First Search) avec Backtracking.

- ☐ **Coder la fonction de vérification** : Écrire une fonction `estValide(x, y)` qui vérifie si une case est dans les limites et n'a pas encore été visitée.
- ☐ **Implémenter le Backtracking** : Écrire la fonction récursive principale qui :
 - Marque la case actuelle comme visitée.
 - Vérifie la condition d'arrêt (compteur = taille totale).
 - Teste récursivement les mouvements possibles.
 - **Backtrack** : Remet la case à 0 (non visitée) si le chemin mène à une impasse.
- ☐ **Test initial** : Valider l'algorithme sur un petit échiquier (5×5) pour éviter les temps de calcul trop longs lors du débogage.

Phase 3 : Livrables et Cas Spécifiques

Répondre aux demandes précises du sujet concernant les parcours et les positions.

- ☐ **Cas 1 - Parcours simple (Ouvert)** : Trouver un chemin hamiltonien depuis une case quelconque.
- ☐ **Visualisation** : Créer un affichage (console ou graphique) dessinant le chemin sur l'échiquier.
- ☐ **La Figure 3** : Configurer le point de départ comme sur la figure 3 du sujet et trouver la solution.
- ☐ **Cas 2 - Le Tour Fermé (Cycle)** : Modifier l'algorithme pour que la dernière case permette de sauter vers la première.
 - ☐ Appliquer au cas 6×6 .
 - ☐ Appliquer au cas 8×8 .

Phase 4 : Analyse Mathématique et Extension

Prendre du recul théorique sur les résultats.

- ☐ **Analyse des symétries :** Sans relancer le programme, déduire d'autres solutions à partir de celle de la Figure 3 (rotations, miroirs).
- ☐ **Étude des cas impossibles :** Démontrer ou tester l'impossibilité pour les tailles 3×3 et 4×4 .
- ☐ **Généralisation (Bonus) :** Réfléchir à la condition d'existence pour une taille $m \times n$.

Phase 5 : Finalisation du Rendu

- ☐ Vérifier que le code est commenté et lisible.
- ☐ Générer les captures d'écran des solutions (Figures 3 et 4).
- ☐ Rédiger les justifications pour les questions théoriques (symétries, tailles impossibles).