

**Exercice 1.** *Le problème du cavalier.*

Dans le jeu d'échecs, le cavalier peut se déplacer deux cases dans une direction, puis une case dans une direction perpendiculaire. Le problème du cavalier est le suivant : Un cavalier posé sur une case quelconque d'un échiquier de dimension  $n \times n$  doit en visiter toutes les cases sans passer deux fois par la même. Le parcours peut être ouvert, c'est à dire, il n'est pas nécessaire de revenir à la case de départ.

1. Donner une solution possible pour un échiquier de dimension  $n = 5$ .
2. Écrire un algorithme basé sur le backtracking construisant tous les parcours ouverts d'un échiquier de dimension  $n$ .

**Exercice 2.** *Carrés latins.*

Un carré latin d'ordre  $n$  est un tableau carré d'entiers de  $n$  lignes et  $n$  colonnes dont chaque ligne et chaque colonne sont des permutations des entiers  $1, 2, \dots, n$  (c'est-à-dire contiennent chacune exactement une fois chacun de ces nombres). Un tel carré est dit normalisé si la première ligne et la première colonne contiennent ces  $n$  nombres dans l'ordre (permutation identité).

- 1a. Donner les quatre carrés latins normalisés de côté  $n = 4$ .
- 1b. Développer la suite des essais que réalisera typiquement un algorithme de backtracking pour  $n = 4$ .
2. Écrire un algorithme basé sur le backtracking construisant tous les tableaux normalisés de côté  $n$ .

**Exercice 3.** *Mots sans carré.*

Un mot est dit *sans carré* s'il ne contient pas deux motifs égaux consécutifs. Exemples :

- $AABC$  ou  $ABCBC$  contiennent des carrés,
- $ABC$  ou  $ABCBACAB$  sont sans carré.

Construire un mot de longueur 100 sur l'alphabet  $\{A, B, C\}$  qui soit sans carré.