

Systemes intelligents

TP 1 : Algorithmes gloutons

Wahabou ABDOUN
2020 - 2021

On souhaite placer un maximum de reines sur un échiquier de taille $n \times n$. Les reines doivent être placées de manière à ce qu'elles ne soient pas en prises (aucune reine n'est menacée par une autre). Ce implique qu'on ne doit pas avoir deux reines situées sur une même ligne, colonne ou diagonale.

Pour ce problème, une solution possible n'est pas simplement le nombre de reines qu'on pourrait placer, mais plutôt les coordonnées de chacune d'elles.

Exercice 1 : Programmes de base

Un échiquier sera décrit par une classe `Echiquier` qui contient une matrice carrée de taille n dont les objets sont de type `Cellule`.

1. Soit la classe `Cellule` définie comme suit :

```
public class Cellule {  
  
    private int x;  
    private int y;  
  
    private int typeOccupation; //libre, reine, menacée par une reine  
  
    final static int LIBRE = 0;  
    final static int REINE = 1;  
    final static int MENACEE = 2;  
  
    public Cellule (int x, int y) {  
        this.x = x;  
        this.y = y;  
        this.typeOccupation = LIBRE;  
    }  
  
    ....  
    ....  
}
```

Complétez cette classe en définissant des accesseurs (*getters*) et des mutateurs (*setters*).

2. Soit la classe `Echiquier` définie comme suit :

```
public class Echiquier {  
  
    private Cellule[][] echiquier;  
    private int taille;  
  
    public Echiquier(int taille) {  
  
        this.taille = taille;  
        echiquier = new Cellule[taille][taille];  
        initialiserEchiquier();  
    }  
  
    public void initialiserEchiquier() {  
  
        for (int x = 0; x < taille; x++) {  
            for (int y = 0; y < taille; y++) {  
                echiquier[x][y] = new Cellule(x, y);  
            }  
        }  
    }  
  
    public void modifierCellule(int x, int y, int valeur) {  
  
        echiquier[x][y].setTypeOccupation(valeur);  
    }  
  
    ....  
    ....  
}
```

Complétez cette classe avec une méthode `placerReine(int x, int y)` qui place une reine dans la cellule de coordonnées `x` et `y` si celle-ci est libre. Ensuite, la méthode modifiera le type d'occupation de toutes les cellules menacées par la reine nouvellement placée (sur les lignes, colonnes et diagonales) comme l'illustre la figure 1.

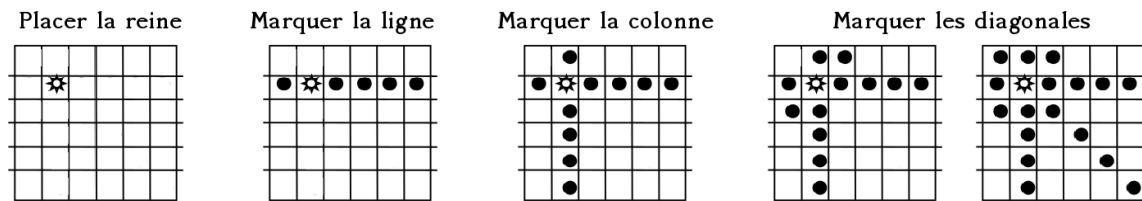


Figure 1 – Placement d'une reine et marquage des cellules menacées

3. Dans la classe `Echiquier`, ajoutez une méthode `toString` qui faciliterait l'affichage de votre échiquier (notamment l'occupation des cellules).
4. Définissez une classe contenant une méthode `main` permettant de tester le placement d'une reine.

Exercice 2 : Algorithme de placement de reines

1. On définit l'algorithme suivant :
 - a) choisir la cellule où la reine peut être placée de manière à menacer le moins de cellules possibles ;
 - b) marquer toutes les cellules menacées ;
 - c) reprendre la première étape (en retirant du décompte toutes les cellules qui étaient déjà menacées) jusqu'à ce qu'il n'y ait plus de cellule libre ;
 - d) donner le nombre reines ainsi que leurs positions.

Implémentez cet algorithme.

2. Proposez un nouvel algorithme de placement de reines. Faites-le valider par votre idée par l'enseignant puis implémentez votre méthode et comparez-la l'algorithme de la question 1.