

Université de Bejaia.
Faculté des Sciences Exactes.
Département Informatique

Travaux Pratiques
Algorithmique Avancée :
Mini-Projets

Master 1
RN – Réseaux et Sécurité

L'objectif principal.

L'objectif de ce mini-projet est de mettre en pratique les concepts d'Algorithmique avancée abordés en cours, notamment :

- La mise en œuvre de structures de données ;
- L'utilisation de la récursivité et des paradigmes étudiés tels que le paradigme "Diviser pour régner"

Ceci afin de les appliquer à la résolution de problèmes connus. Trois mini-projets sont proposés dans ce qui suit.

Sujet N°1 : Rush Hour

Le jeu simule une congestion automobile dans un parking à l'heure de pointe (appelée « rush hour » en anglais). Le but du jeu est d'extraire le véhicule rouge d'une grille dans laquelle plusieurs autres véhicules bloquent la sortie. Il faut pour cela les déplacer, mais ils sont suffisamment entrecroisés et il y a suffisamment de contraintes de déplacement pour que la solution ne soit pas triviale. Le joueur doit déplacer les autres véhicules afin d'amener graduellement le véhicule rouge vers la sortie. La grille impose la direction de déplacement des divers véhicules. Le joueur n'a pas le droit de soulever les véhicules et ne peut les déplacer que dans les directions imposées par la grille de jeu.

Ecrire un programme JAVA qui résout efficacement le jeu pour un niveau quelconque donné.

Sujet N°2 : les deux points les plus proches dans un plan

Le défi consiste à identifier la paire de points la plus proche dans un plan, un problème crucial avec de nombreuses applications. Par exemple, dans le domaine du contrôle du trafic aérien, il est essentiel de détecter les avions qui se rapprochent trop près les uns des autres, car cela pourrait signaler une collision imminente.

Ecrire un programme JAVA efficace qui, pour n différents points données par leurs abscisses et ordonnées, trouve la paire la plus proche.

Sujet N°3 : Le problème des n reines

Le problème des n reines consiste à placer n reines sur un échiquier de taille $n \times n$ de manière à ce qu'aucune reine ne soit menacée par une autre, c'est-à-dire qu'il ne doit pas y avoir plus d'une reine par ligne, par colonne ou par diagonale. L'objectif de cet exercice est d'écrire un programme efficace en Java qui calcule le nombre de façons différentes de placer les n reines sur l'échiquier sans qu'elles se menacent mutuellement.