Sujet n°3bis

Algorithmique & Programmation

Problèmes algorithmiques récursifs

Ce sujet de TP est un complément pour ceux qui ont terminé le TP 3. Avant de commencer ce sujet, merci de vérifier que vous avez bien réalisé correctement tous les exercices du TP3. N'hésitez à pas à demander aux enseignants de valider les différents algorithmes que vous avez implémentés en Python.

Exercice 1 : *Labyrinthe (récursivité)*

Dans cet exercice, nous vous proposons de programmer un algorithme qui permettra de faire sortir un personnage d'un labyrinthe. Notre labyrinthe sera représenté sous la forme d'un tableau à deux dimensions (en Python, une liste de listes). Une valeur de 0 représentera un espace sur lequel notre personnage peut marcher, une valeur de 1 représentera un mur. La sortie du labyrinthe aura la valeur 2. Enfin, notre personnage sera lui représenté par le chiffre 3. La déclaration initiale de ce tableau devrait donc ressembler à ceci :

```
labyrinthe=[ [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [ 1, 3, 1, 0, 0, 0, 0, 0, 0, 1],
      [ 1, 0, 1, 1, 1, 1, 0, 1, 0, 1],
      [ 1, 0, 0, 0, 1, 1, 0, 1, 0, 1],
      [ 1, 0, 1, 1, 1, 1, 0, 1, 0, 1],
      [ 1, 0, 1, 0, 1, 1, 0, 1, 0, 2],
      [ 1, 0, 1, 0, 0, 0, 0, 1, 1, 1],
      [ 1, 0, 1, 0, 1, 1, 0, 0, 0, 1],
      [ 1, 0, 0, 0, 1, 1, 0, 1, 0, 1],
      [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
```

Question 1. Créez un algorithme récursif permettant de sortir de ce labyrinthe.

Question 2. (Optionnel) Transformez cet algorithme en une version itérative. Quelle version est la plus simple ?

Exercice 2 : *Comparaison d'algorithmes de tri*

Vous avez mis en œuvre une version récursive du tri rapide (quick-sort) dans le TP3. Dans cet exercice nous vous proposons d'implémenter quelques autres algorithmes de tri et de comparer leur vitesse d'exécution.

Question 1. Créez un programme permettant de calculer la vitesse d'exécution de votre implémentation du tri-rapide pour différentes tailles de tableau (10, 100, 1000 et 10000 éléments) initialisés avec des valeurs aléatoires (‘random‘). Vous pourrez utiliser la fonction ‘clock()‘ du module ‘time‘ pour mesurer le temps avec précision.

Question 2. Modifier ce programme afin d'effectuer ces mêmes tests sur des tableaux déjà triés en ordre ascendant et descendant.

Question 3. Implémentez le tri par sélection et testez le.

Question 4. Implémentez le tri par insertion et testez le.

Question 5. Implémentez le tri à bulle et testez le.

Question 6. Quelles sont vos conclusions ?