TP : Cannibales &Missionnaires

Algorithmes de recherche en profondeur

Réalisé par:

ERRAZI Fatima-Zahra

AOUANET Bahaeddine

Introduction

le jeu des missionnaires et des cannibales consiste à trouver une série de traversées dans une barque,

limitée à deux individus maximum, pour déplacer un groupe de trois missionnaires et trois cannibales d'unerive à l'autre d'un fleuve.

Le but du jeu :

Le but est de passer de la situation initiale où les six personnages se trouvent sur la rive droite à une situation finale où ils sont tous sur la rive gauche,

tout en veillant à ce que les cannibales ne mettent pas en danger la vie des missionnaires à aucun moment durant les traversées. Le défi réside dans la planification stratégique des déplacements pour assurer la sécurité des missionnaires tout au long du jeu.

Modélisation du jeu

Pour modéliser l'espace d'états, dans notre cas,

chaque état spécifie le nombre de missionnaires et de cannibales restant sur la rive droite,ainsi que la position de la barque, qu'elle soit à droite ou à gauche du fleuve.Cet espace d'états est représenté par des triplets (R, x, y), où R désigne la position de la barque (D pour la rive droite, G pour la rive gauche),

x représente le nombre de missionnaires restant sur la rive droite, et y indique le nombre de cannibales restant sur la rive droite. Les valeurs possibles pour x et y sont comprises entre 0 et 3 inclus. Pour assurer la sécurité des missionnaires, certaines contraintes doivent être respectées : soit il n'y a plus de missionnaires sur la rive droite (x = 0), soit le nombre de missionnaires restant est supérieur ou égal au nombre de cannibales restant (x ≥ y).

Les règles du jeu

Le jeu présente un ensemble de règles régissant les déplacements des missionnaires et des cannibales.

Cela inclut la restriction selon laquelle seulement deux individus peuvent être transportés dans la barque au maximum parmi les trois missionnaires et les trois cannibales disponibles.

R10 : 1 missionnaire , 0 cannibale

R01 : 0 missionnaire , 1 cannibale

R02 : 0 missionnaire, 2 cannibales

R20 : 2 missionnaires, 0 cannibale

R11 : 1 missionnaire, 1 cannibal

La recherche en profondeur

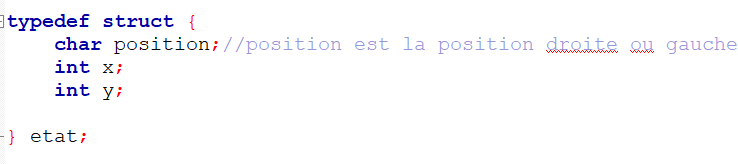
La recherche en profondeur (une recherche systématique) consiste à explorer une branche de l'arbre de recherche jusqu'à trouver l'état final ou à faire marche arrière pour explorer d'autres possibilités.

Cette stratégie utilise une pile pour stocker les nœuds générés, avec des successeurs ajoutés au début de la liste pour l'exploration.

La pile est contrôle par son sommet.

Implémentation

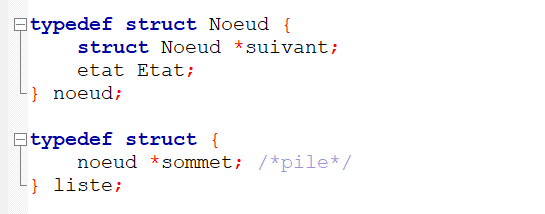
* Le triplet est défini par la structure *etat* suivante:



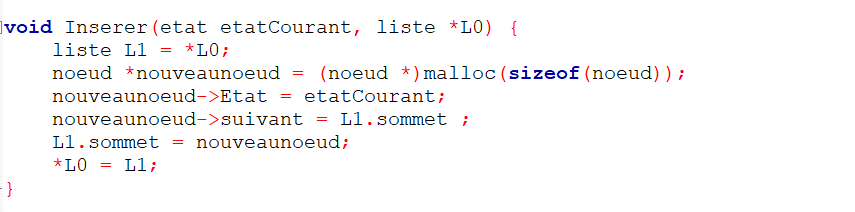
Tel que:

|  |  |
| --- | --- |
| variable | signification |
| position | position de la barque |
| x | nombre de missionnaires sur la rive droite |
| y | nombre de missionnaires sur la rive gauche |

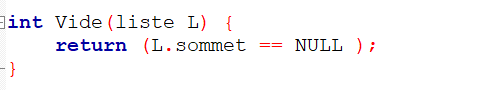
* La structure noeud définit les éléments de la pile:
* suivant pointeur sur l’élément suivant de la pile.
* Etat qui définit l’état du noeud de l’arbre de type la structure etat définie ci-dessus.
* La structure *liste* définit la pile avec un pointeur sommet.

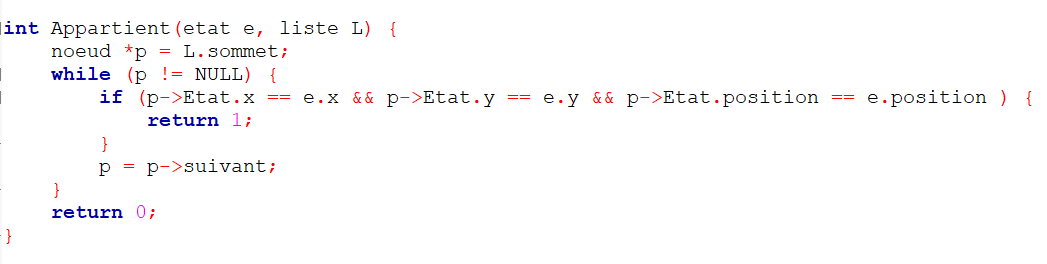


* La fonction *Inserer* permet d’ajouter les successeurs des nœuds qui seront générés à la pile

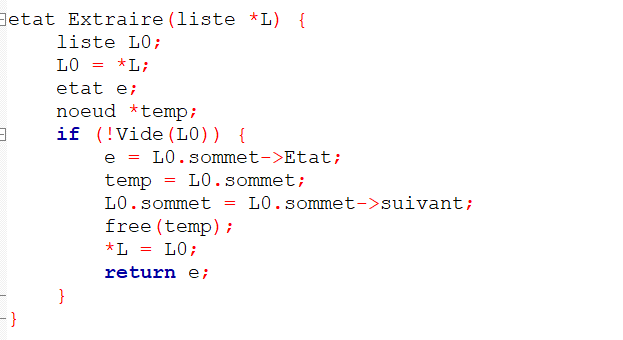


* Les fonctions *Vider* et *Appartient* testent respectivement si la liste de travail est vide et si un noeud y appartient ou non.

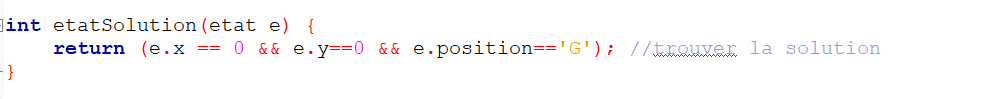




* La fonction *Extraire* permet d’extraire le noeud à traiter et le retourne



* La fonction *etatSolution* vérifie si le noeud extrait correspond-t-il à l’état final attendu ou non



* La fonction *RechercheProfondeur* est la fonction qui fait appel à toutes les fonctions définie pour chercher la solution selon le schèma suivant:

État initial

Pas solution

solution

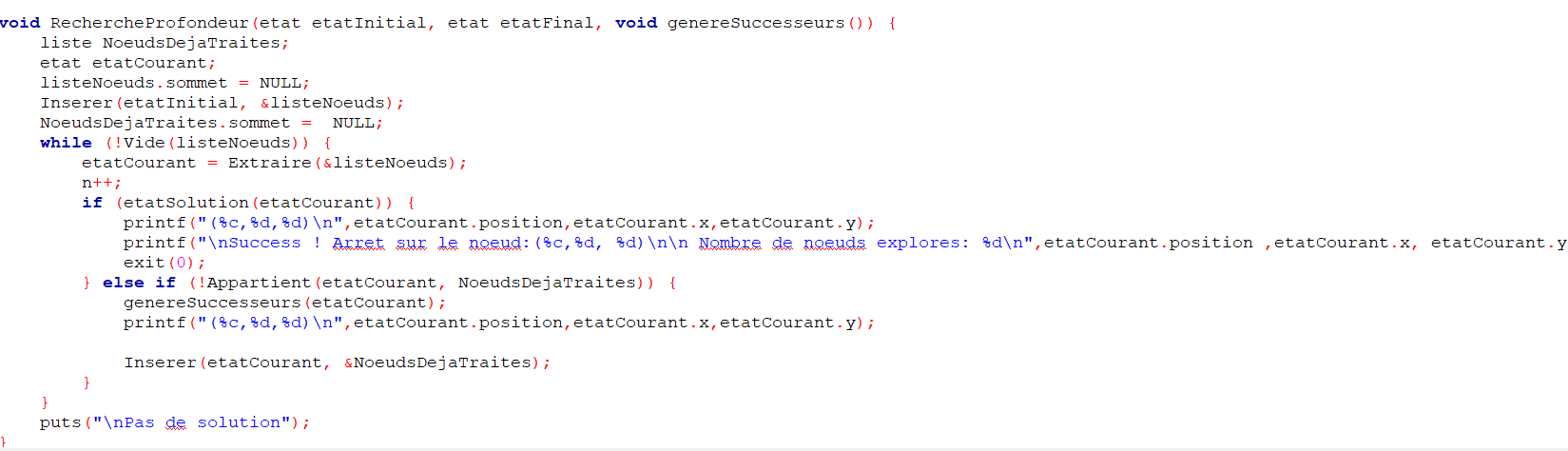
Générer successeurs

S’arreter

Pas solution

Extraire Fils du noeud courant extrait

Solution



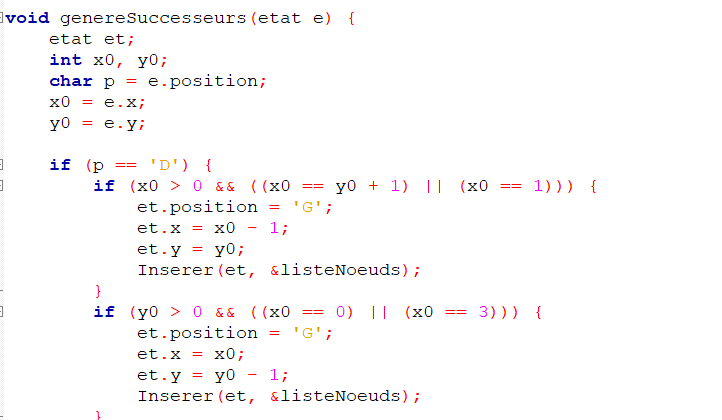
* La fonction *genereSuccesseurs* est codée selon les règles du jeu. Elle permet de générer les nœuds fils d’un état.

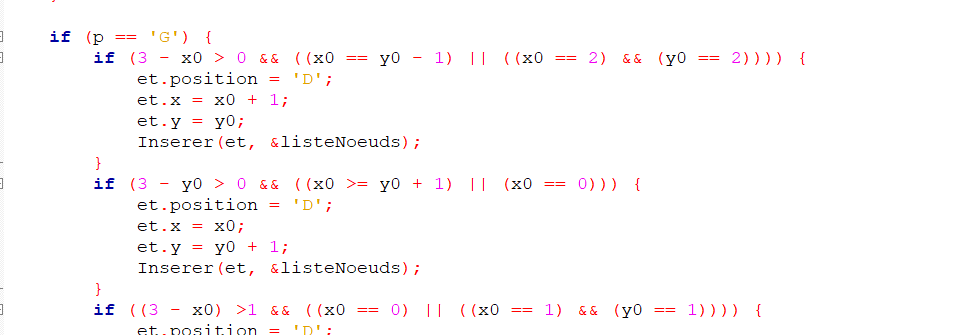
Pour l’implémenter on a divisé le code en deux blocs *if:*

* Bloc qui permet de générer les états si la barque était à droite (position=’D’) et doit se déplacer à gauche.
* Bloc qui permet de générer les états si la barque était à gauche(position=’G’) et doit se déplacer à droite.

Sachant que la barque doit au minimum être conduite par un missionnaire ou un cannibale et ne comporte que deux .

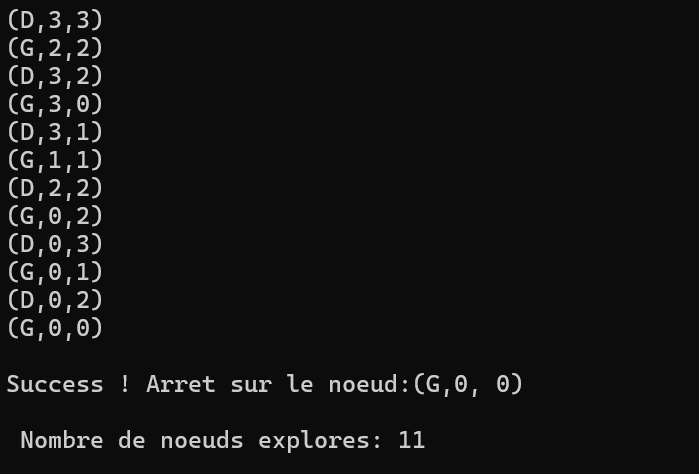
Des extraits de la fonctions sont les suivants(cf. Code source):





Exécution

L’exécution de la fonction donne le résultat suivant:



* Pour arriver à l’état final, il a eu le parcours de 11 nœuds.
* Les nœuds affichés sont le chemin pour trouver l’état recherché.

L’arbre de recherche

(D,0,3)

(G,2,2)

(D,3,2)

(D,3,3)

(G,3,0)

(D,3,1)

(D,2,3)

(G,3,2)

(D,3,3)

(D,3,3)

(G,0,2)

(D,2,2)

(G,1,1)

(G,0,0)

(D,0,2)

(G,0,1)

(G,3,1)

(D,3,3)