



Projet d'Informatique Scientifique

Etudiante: Elizabeth Gandibleux

Groupe: 681C

Encadrant : **Prof. Dr. Hab. Xavier Gandibleux**

1 Partie 1

1.1 Execution du code

Les instances *arena.map* et *didactic.map* ont été choisies pour tester le programme. Par défaut, le point de départ est: **(5,4)** et le point d'arrivée est **(10,7)**.

Dans un premier temps il faudra sélectionner l'instance voulue

Instances numériques à choisir:

```
arena1.map  
> didactic.map
```

Ici nous sélectionnons *didactic.map*.

Dans un second temps, il faudra choisir l'algorithme qui est choisit pour être exécuté ou alors demander au programme d'exécuté tous les algorithmes

Algorithme(s) à choisir:

```
Dijkstra  
A*  
> Tous
```

1.2 Approche "recherche informée optimale": *Algorithme de dijkstra*

1.2.1 Structures choisies

- Matrice de caracteres: **A**
- Matrice d'entiers: **matriceNumerique**
- Matrice de tuples d'entiers: **matriceOriginelle**
- Une file de priorité, autrement dit un tas: **pQ**
- File de tuples d'entiers: **listeFinale**
- Booléen : **trouve**

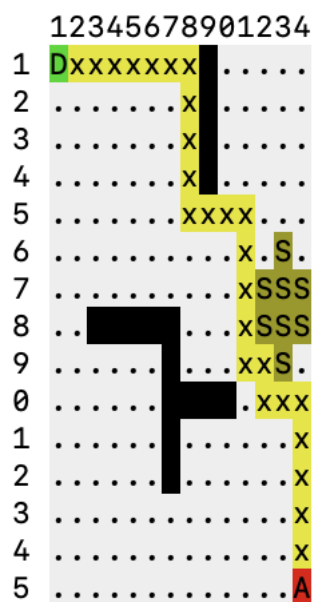
1.2.2 Jeux de test

Instance choisie: didactic.map

D::Tuple{Int64,Int64} = (1,1)

Ar::Tuple{Int64,Int64} = (15,14)

Dijkstra Algorithmme ...



Distance D → A: 27

Number of states evaluated: 193

Dijkstra fait ...

1.3 Approche "recherche informée heuristique": *Algorithme du A**

1.3.1 Structures choisies

Les structures choisies sont les mêmes que celles dans l'algorithme de Dijkstra

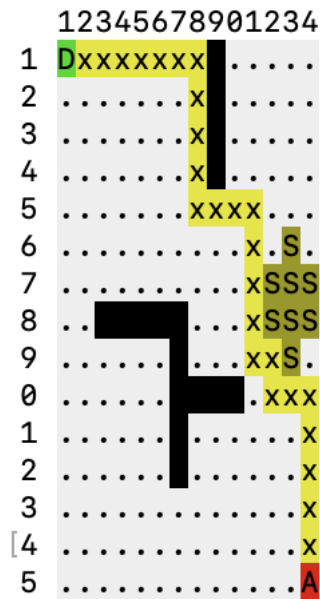
Instance choisie: didactic.map

D::Tuple{Int64,Int64} = (1,1)

Ar::Tuple{Int64,Int64} = (15,14)

1.3.2 Jeux de test

A* Algorithmme ...



Distance D → A: 27

Number of states evaluated: 173

A* fait ...

2 Partie 2

3 Conclusion