

Projet PATIA - Résolution de Problèmes

1. Recherche dans un espace d'états (Taquin)

Description

Implémentation des algorithmes de recherche :

- BFS (Breadth-First Search)
- DFS (Depth-First Search)
- A* (A-Star)

Structure

```
taquin/  
├── benchmarks/ # Benchmarks générés  
├── scripts/  
│   ├── generate_npuzzle.py  
│   ├── solve_npuzzle.py  
│   └── performance_analysis.ipynb  
└── README.md
```

Utilisation

Génération des benchmarks :

```
python scripts/generate_npuzzle.py -s 3 -ml 5 -n 10 benchmarks -v
```

Résolution avec algo bfs :

```
python scripts/solve_npuzzle.py benchmarks/npuzzle_3x3_len1_0.txt -a bfs -v
```

Résolution avec algo dfs :

```
python scripts/solve_npuzzle.py benchmarks/npuzzle_3x3_len1_0.txt -a dfs -v
```

Résolution avec algo astar :

```
python scripts/solve_npuzzle.py benchmarks/npuzzle_3x3_len1_0.txt -a astar -v
```

Analyse des performances

```
jupyter notebook scripts/performance_analysis.ipynb
```

Encodage

- Chaque tuile est représentée par une lettre (A, B, C...)
- La case vide est représentée par 0

- Format : A 0 B C D E F G H pour un 3x3

Résultats

- Graphiques comparant les temps d'exécution
- Analyse de la complexité en fonction de la taille