

20 - 2 인공지능 과제 1

미로 탈출하기

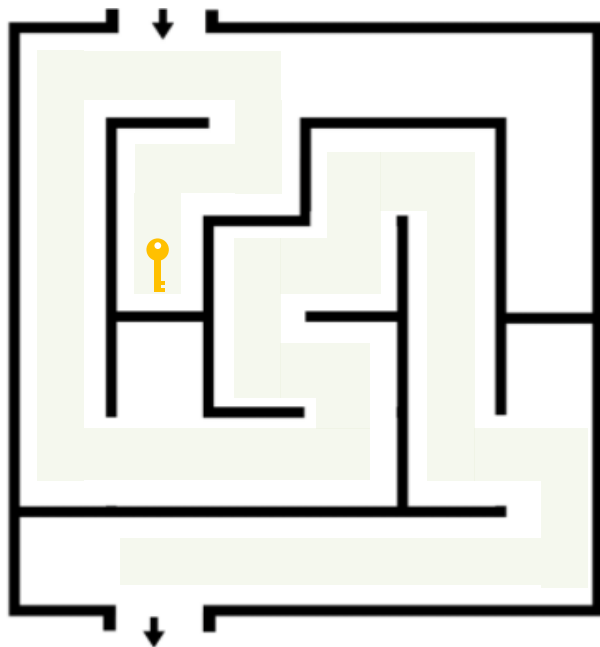
조교 이봉석, 조충현
tjr4090@hanyang.ac.kr
johnart@hanyang.ac.kr

과제 개요

- 다양한 Search Algorithm을 이용하여 미로 경로 탐색.
- 세부사항
 - 총 **4개의 미로**를 탈출하는 것이 목표
 - 미로에는 **입구**와 **출구**가 있으며, **통로**와 **벽**이 구분되어 있음.
 - 출구는 자물쇠로 잠겨 있어 출구로 가는 도중에 **키를 습득**해야 함.
 - 키가 여러 개인 경우 **모두 습득**해야 하며 습득 순서는 **무관함**.
 - 알고리즘 : Breadth first search, Iterative deepening search,
Greedy best first search, A* algorithm
 - 각 미로마다 위에 적힌 **네 가지의 알고리즘**을 모두 사용.
 - 각각의 미로를 탈출하기 위해 **알고리즘 별로 탐색한 노드의 개수와 최단 경로(거리)**로 점수 산출.

결과물 예시

- 출력 : 1,2,3,4,5 로 표현된 m*n 행렬 , 최단 경로 길이, 탐색 노드 개수
(1 = 벽, 2 = 통로, 3 = 출발점, 4 = 도착점, 5 = 최적 탐색 경로)



1	1	③	1	1	1	1	1	1	1	1
1	5	5	5	5	2	2	2	2	2	1
1	5	1	1	5	1	1	1	1	2	1
1	5	1	5	5	1	5	5	5	1	2
1	5	1	5	1	5	5	1	5	1	2
1	5	1	5	1	5	1	1	5	1	1
1	5	1	1	1	5	5	1	5	1	2
1	5	1	2	1	1	5	1	5	1	2
1	5	5	5	5	5	5	1	5	5	5
1	1	1	1	1	1	1	1	1	5	1
1	2	5	5	5	5	5	5	5	5	5
1	1	④	1	1	1	1	1	1	1	1

length=55
time=74

점수 산출 - 코드

- 코드 (70%)

- 파일 이름

- 본인학번_assignment_1.py (파이썬 파일 **한 개 로만 작동** 가능 하도록)

- 구조

- BFS – `def bfs(...)`
 - IDS – `def ids(...)`
 - GBFS – `def gbfs(...)`
 - A* – `def a_star(...)`



함수명은 변경 불가
**실행 후 각 미로에 대해 알고리즘마다
Maze_N*_output.txt 생성**

- 필요에 따라 작성한 기타 함수들 (N은 N번째 미로, *는 BFS, IDS, GBFS, A_star)

※ 함수들의 인자와 리턴 값은 자유 (단 보고서에 명시)

- Test

- 임의의 미로를 이용하여 실제 작동 여부를 평가하여 점수 산출

점수 산출 - 보고서

- **보고서 (30%)**
 - **코드 설명**
 - 각 미로마다 **코드 실행방법**을 명시. (5%)
 - 각 미로마다 **코드 동작**에 대해 설명. (5%)
 - **함수 설명**
 - 각 **함수마다 어떻게 짰는지/사용 용도**에 대한 설명. (10%)
 - **실험결과**
 - **최단경로(length)** (5%)
 - True / False (**최단경로가 맞는지** 확인하여 점수 산출)
 - **탐색한 노드의 개수(time)** (5%)
 - 미로와, 이동 방향의 우선순위에 따라 달라질 수 있음.
 - 각 미로의 알고리즘마다 학생 전체 평균과의 차이를 계산하여 점수 산출

과제 조건

- 환경

- 프로그래밍 언어 : **Python 3.x**
- OS : **Windows**
- 외부 라이브러리 사용 **불가**(내장함수만 사용 가능)
- 보고서 : **MS Word**

- 제출 사항

- 파이썬 파일 : 본인학번_assignment_1.py (파이썬 **파일 한 개**) (70%)
- 결과 보고서 : 본인학번_assignment_1.docx
 - 코드 설명 (10%)
 - 함수 설명 (10%)
 - 실험 결과 (10%)
- 출력파일 (16개) : **Maze_N*_output.txt** (N은 N번째 미로, *는 BFS, IDS, GBFS, A_star)

주의 사항

- 파일명 반드시 준수.
- 파일은 **Blackboard**에 올려주세요.
- 제출 기한 : **2020.10.11 (23:59)**
- 추가 제출 기한 **없음**.
- 점수 비중 : **코드 70% 보고서 30%**

주의 사항

- 아래 예시와 같은 입출력 형식을 반드시 준수.
- 코드와 같은 경로에 input.txt와 output.txt 관리.

입력 input.txt 에서 read

```
1 12 12
1 1 3 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2 1
1 2 1 1 2 1 1 1 1 1 2 1
1 2 1 2 2 1 2 2 2 1 2 1
1 2 1 2 1 2 2 1 2 1 2 1
1 2 1 6 1 2 1 1 2 1 1 1
1 2 1 1 1 2 2 1 2 1 2 1
1 2 1 2 1 1 2 1 2 1 2 1
1 2 2 2 2 2 2 1 2 2 2 1
1 1 1 1 1 1 1 1 1 1 2 1
1 2 2 2 2 2 2 2 2 2 2 1
1 1 4 1 1 1 1 1 1 1 1 1
```

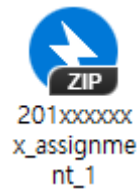
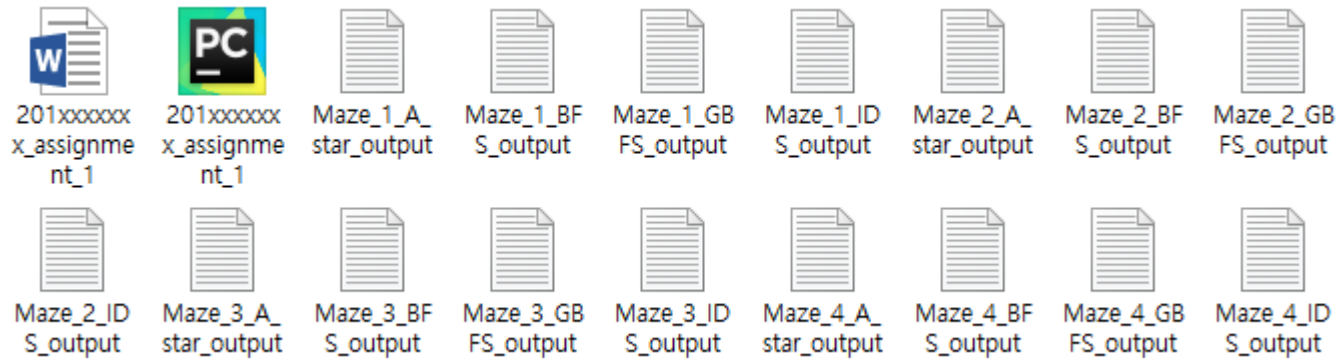
출력 output.txt 로 write

```
1 1 3 1 1 1 1 1 1 1 1 1
1 5 5 5 5 2 2 2 2 2 2 1
1 5 1 1 5 1 1 1 1 1 2 1
1 5 1 5 5 1 5 5 5 1 2 1
1 5 1 5 1 5 5 1 5 1 2 1
1 5 1 5 1 5 1 1 5 1 1 1
1 5 1 1 1 5 5 1 5 1 2 1
1 5 1 2 1 1 5 1 5 1 2 1
1 5 5 5 5 5 5 1 5 5 5 1
1 1 1 1 1 1 1 1 1 1 5 1
1 2 5 5 5 5 5 5 5 5 5 1
1 1 4 1 1 1 1 1 1 1 1 1
---
```

length=55
time=74

주의 사항

- 제출하는 파일들을 압축하여 BlackBoard에 올릴 것!



Thank you!
