

# 인공지능

## Programming #1

C184036 이소연

제출일 : 10월 13일 (금)

▪ maze1

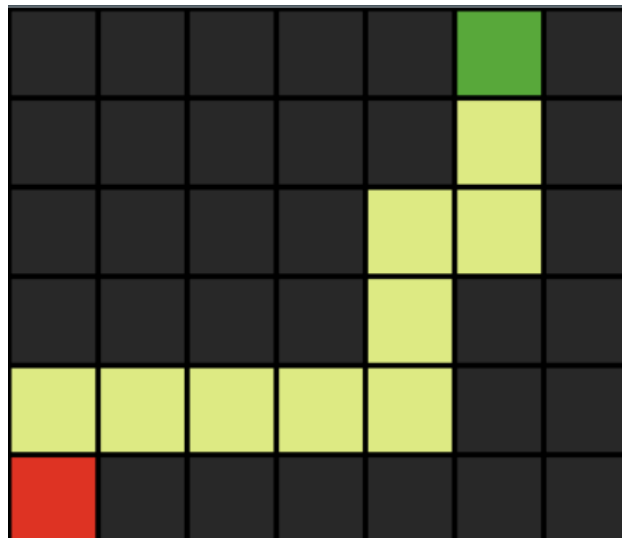
<DFS>

Maze:

```
?????????B??
????????? ??
???????? ??
???????? ????
      ???
A??????????
```

Solving...  
States Explored: 11  
Solution:

```
?????????B??
?????????*??
????????**??
????????*????
*****????
A??????????
```



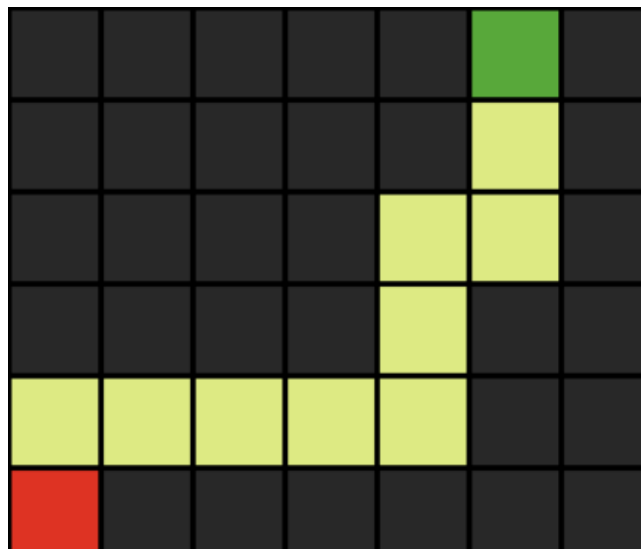
<BFS>

Maze:

```
?????????B??
????????? ??
???????? ??
???????? ????
      ???
A??????????
```

Solving...  
States Explored: 11  
Solution:

```
?????????B??
?????????*??
????????**??
????????*????
*****????
A??????????
```



maze1 비교 결과 : DFS 알고리즘과 BFS 알고리즘의 차이가 없는 것을 확인할 수 있다.

<DFS>

Maze:

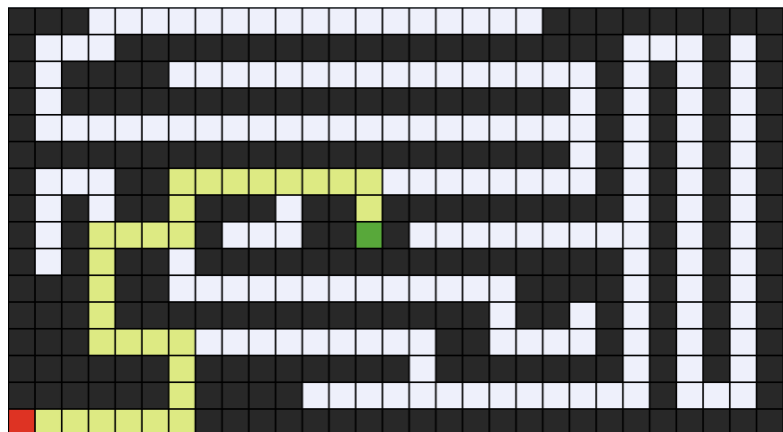
Figure 1: Schematic representation of the 1000 Genomes Project. The diagram shows a large circle representing the human genome, divided into segments. A smaller circle inside represents a specific individual's genome. The segments are labeled with 'A' and 'B' to indicate different populations or individuals. The diagram illustrates the relationship between the 1000 Genomes Project and the study of genetic variation.

Solving...  
States Explored: 194  
Solution:

```

TTTTTTT                                TTTTTTTTTTTTTTTTTTTT
TT  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT
TT  TTTTTTTT                                TT  TT  TT
TT  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT  TT
TT                                TT  TT  TT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT  TT
TT  TTTT***** TT  TT  TT
TT  TT*TTTTT*TTTTT*TTTTTTTTTTTTTTTTTTT TT  TT
TT  TT***** TTTTBT TT  TT
TT  TT*TTTT TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT
TTTTTT*TTTT TTTTTTTT TT  TT
TTTTTT*TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT  TT
TTTTTT*TTT TT  TT  TT
TTTTTT*TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT  TT
TTTTTT*TTT TT  TT  TT
TTTTTT*TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT TT  TT  TT
TTTTTT*TTTTTTTTTT TT  TT
A*****TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

```



<BFS>

Maze:

```

?????              ?????????????????
??  ???????????????????????????????????  ??  ??
??  ??????????              ??  ??  ??
??  ???????????????????????????????????????????????  ??  ??  ??
??              ??  ??  ??  ??
??????????????????????????????????????????????????  ??  ??  ??
??  ?????              ??  ??  ??
??  ??  ??  ???????  ???????????????????????????  ??  ??
??  ??  ??  ???????              ??  ??  ??
??  ??  ?????  ???????????????????????????????????????  ??  ??
??????  ?????              ???????????  ??  ??  ??
??????  ???????????????????????????????????????????  ??  ??  ??
??????              ?????  ??  ??  ??  ??
??????????????  ???????????????????  ???????????????????  ??  ??
??????????????  ?????????              ??  ??
A  ?????????????????????????????????????????????????

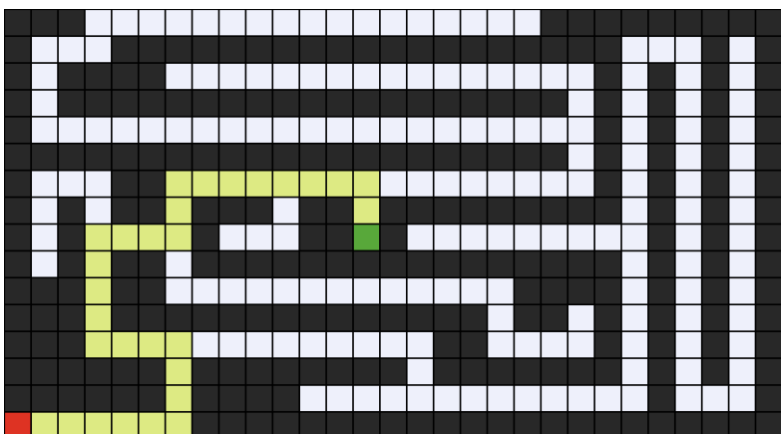
```

Solving...  
States Explored: 77  
Solution:

```

????? ?????????????????
?? ??????????????????????????????????????????? ?? ??
?? ??????????? ?? ?? ?? ??
?? ??????????????????????????????????????????????? ?? ?? ?? ??
?? ?? ?? ?? ??
?????????????????????????????????????????????????????????????????? ?? ?? ?? ??
?? ??????????????????????????????????????????????????????????????????? ?? ?? ?? ??
?? ?? ??????????????????????????????????????????????????????????????????? ?? ??
?? ??+????? ??????????????????????????????????????????????????????????????????? ??
?? ??+????? ???????????????????????????????????????????????????????????????????
??+????+????? ???????????????????????????????????????????????????????????????????
?????+????? ???????????????????????????????????????????????????????????????????
?????+????? ???????????????????????????????????????????????????????????????????
?????+????? ???????????????????????????????????????????????????????????????????
?????+????? ???????????????????????????????????????????????????????????????????
A+?????+?????+?????+?????+?????+?????+?????+?????+?????+?????+?????+?????+?????+

```



maze2 비교 결과 : DFS 알고리즘보다 BFS 알고리즘의 탐색 횟수가 더 적은 것을 알 수 있다. 탐색한 경로가 같다.

▪ maze3

<DFS>

```

Maze:

????  ??
????  ??? ?
??B ??  ??
??  ??? ????
      ????
A???????????

Solving...
States Explored: 17
Solution:

????****??
????*????*??
??B*??*??
??  ?????????
*****????
A???????????

```

<BFS>

```

Maze:

????  ??
????  ??? ?
??B ??  ??
??  ??? ????
      ????
A???????????

Solving...
States Explored: 6
Solution:

????  ??
????  ??? ?
??B ??  ??
??*???? ????
**  ????
A???????????

```

maze3 비교 결과 : DFS 알고리즘보다 BFS 알고리즘의 탐색 횟수가 더 적은 것을 알 수 있다. 탐색한 경로가 서로 다르다.

▪ maze4

<DFS>

```

Maze:

??
?? ??????????
??  ??
???? ?? ??
???? ?? ?? B
?? ??
A ?? ?????
?? ??

Solving...
States Explored: 22
Solution:

??
?? ??????????
?? ***??
????*??* ??
????*??*?? B
??*??*???
A*?? ?????
?? ??

```

<BFS>

```

Maze:

??
?? ??????????
??  ??
???? ?? ??
???? ?? ?? B
?? ??
A ?? ?????
?? ??

Solving...
States Explored: 34
Solution:

??
?? ??????????
?? ***??
????*??* ??
????*??*??B
??*??*???
A*?? ?????
?? ??

```

maze4 비교 결과 : DFS 알고리즘보다 BFS 알고리즘의 탐색 횟수가 더 많은 것을 알 수 있다. 탐색 결과가 살짝 다르다.

## ▪ maze5

### <DFS>

Maze:

```

      ????  ??
    ????? ????B ???? ??
??  ??  ???? ???? ??
??  ?????? ??  ??
??  ??  ???? ??????? ??
    ????? ??  ??
      ??  ??
A ??????  ???????

```

Solving...

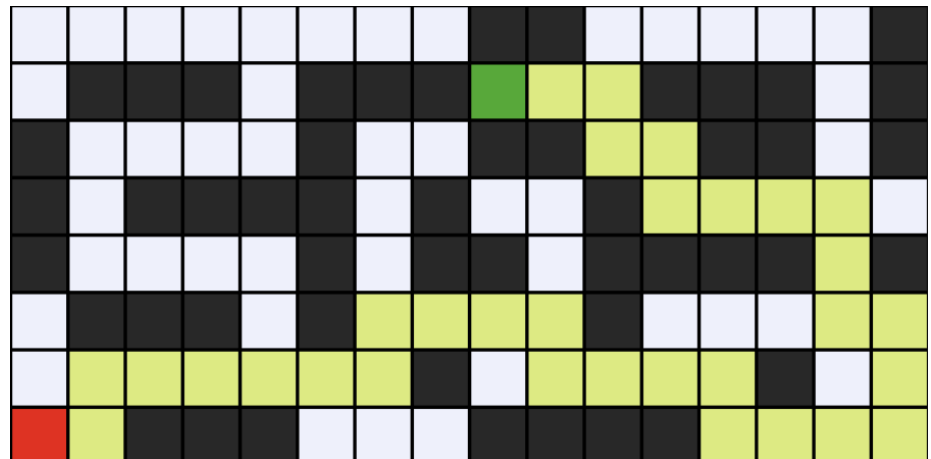
States Explored: 37

Solution:

```

      ????  ??
    ????? ????B**????? ??
??  ??  ?????*????? ??
??  ?????? ??  ??****
??  ??  ???? ????????*??
    ?????? ??****?  **
    *****? *****? *
A*?????  ????????****

```



### <BFS>

Maze:

```

      ????  ??
    ????? ????B ???? ??
??  ??  ???? ???? ??
??  ?????? ??  ??
??  ??  ???? ??????? ??
    ????? ??  ??
      ??  ??
A ??????  ???????

```

Solving...

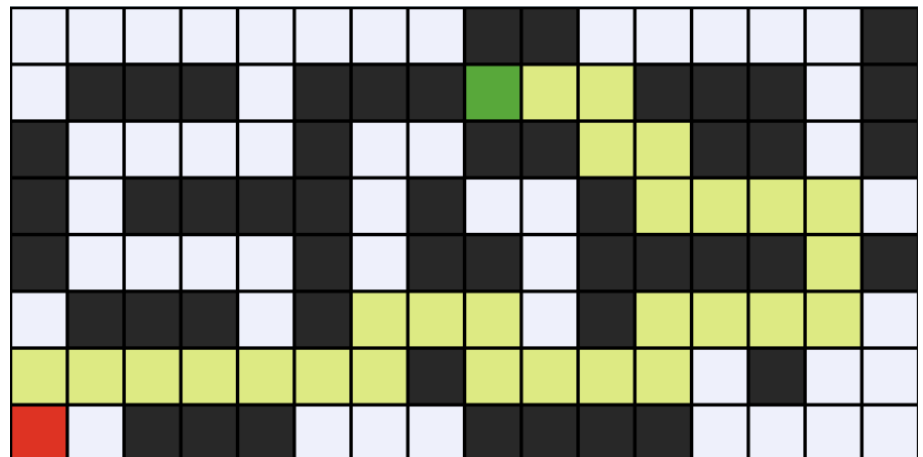
States Explored: 78

Solution:

```

      ????  ??
    ????? ????B*????? ??
??  ??  ?????*????? ??
??  ?????? ??  ??****
??  ??  ???? ????????*??
    ?????? ??***  ??****
    *****?***** ??
A ??????  ???????

```



maze5 비교 결과 : DFS 알고리즘보다 BFS 알고리즘의 탐색 횟수가 더 많은 것을 알 수 있다. 탐색 결과가 살짝 다르다.

#### <총 결과>

- DFS: 목표 지점이 깊은 곳에 있을 경우에 빠를 수 있다. 하지만 최단 경로를 찾는다는 보장이 없으며, 목표 지점에서 멀리 떨어진 노드를 먼저 방문할 가능성이 있다.
- BFS: 최단 경로를 찾는 것이 보장되어 있다. 가까운 노드부터 차례대로 탐색하기 때문에 최초로 발견한 목표 지점이 최단 경로이다.

미로의 구조와 목표에 따라 어떤 탐색 알고리즘이 미로를 해결하는 데 더 효율적인지는 다를 수 있다. 최단 경로를 찾을 때는 BFS를 사용하는 것이 일반적으로 더 좋다.