

Homework 04

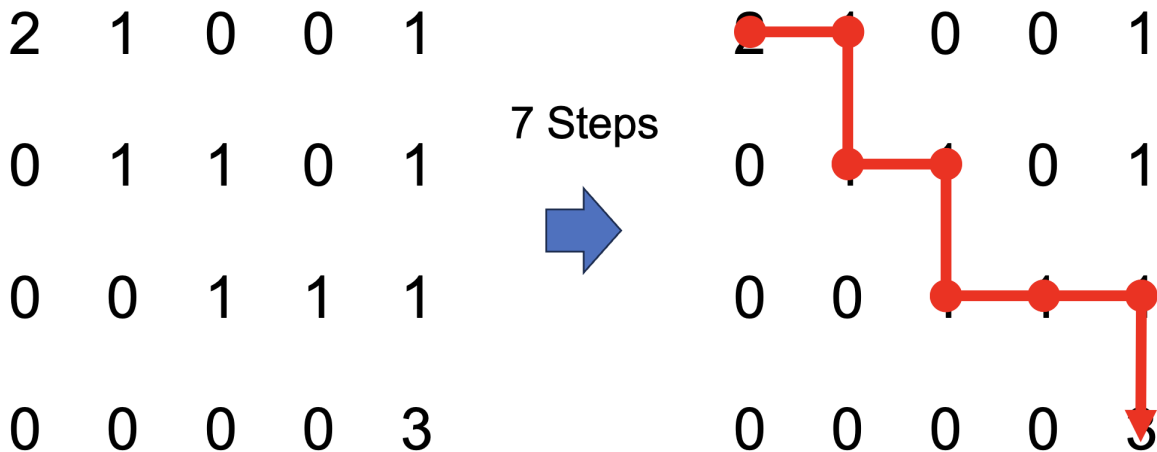
[2023-2] 데이터사이언스를 위한 컴퓨팅의 기초 (M3239.005600 001)

Due: 2023년 11월 24일 금요일 23시 59분

1. Find Maze [30pts]

아래 지시문에 따라 미로의 출구를 찾아 최단거리를 출력하는 코드를 완성하시오.

Instruction:



1. 미로의 크기는 제각각이며 하나뿐인 임의의 지점 2에서 시작해서 1을 지나 하나뿐인 임의의 지점 3에 도착해야 한다. (0은 지나갈 수 없음)
2. 미로는 여러 갈래길이 있을 수 있으나 순환되는 경우는 없다.
3. Maze Class 내 BFS 또는 DFS 어느 것을 구현해도 상관없다.
4. ##### 바깥 영역에 해당하는 코드는 수정이 불가능하다.

Quiz

Q1.1 MakeNode 함수를 완성하시오.

Hint. 한 칸 이동 시 상하좌우로 움직여야 하며 방문한 곳에 또 가지 않아야 한다.

Q1.2 Maze 클래스를 완성하시오.

Hint. 막다른 길은 이동 수에 포함하지 않아야 한다.

Q1.3 FindMaze 함수를 완성하시오.

Hint. 시작 위치를 찾고 방문 여부를 판단할 배열을 만들어야 한다.

Implementation:

```
matrix = [[2, 1, 0, 0, 1],
           [0, 1, 1, 0, 1],
           [0, 0, 1, 1, 1],
           [0, 0, 0, 0, 3]]
FindMaze(matrix)
```

```
matrix = [[1, 1, 0, 3, 1, 1],
          [0, 1, 1, 0, 1, 0],
          [2, 0, 1, 1, 1, 1],
          [1, 0, 0, 1, 0, 1],
          [1, 1, 1, 1, 1, 0]]
FindMaze(matrix)
```

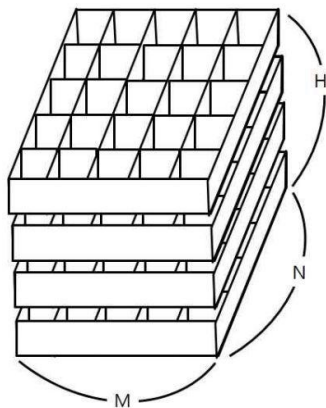
11

2. StealthGuard for all private offices [30pts]

아래 지시문에 따라 모든 직원들이 보안패치(StealthGuard)를 설치할 수 있는지의 여부를 구하는 코드를 완성하십시오.

Instruction:

- (주)컴기전자는 독특한 사무실 구조를 가지고 있다. 전체 건물은 아래의 그림과 같이 격자모양으로 생겼으며, 모든 직원들은 개인 전용 사무실을 가진다.



- 건물 그림에서의 M, N, H는 각각 M(가로길이), N(세로길이), H(높이)이며, 각 범위는 다음과 같다.
($2 \leq M, N \leq 100, 1 \leq H \leq 100$)
- 모든 개인 사무실의 가로, 세로, 높이는 1이다.
- 비어있는 사무실이 있을 수도 있다.
(전체 사무실의 개수 \neq 직원 수)

- 최근, (주)컴기전자는 내부망PC에 대한 보안 강화차원을 위해, 'StealthGuard'라는 보안프로그램을 새로 도입하였다.
 - 'SafeGuard'는 특별한 규칙이 없이 임의로 배포되었다. 현재, 일부 내부망 PC들은 설치가 완료되었으나 아직 설치되지 않은 PC도 많은 상태이다.
 - 'SafeGuard' 프로그램의 설치방식은 다음과 같다.
 - 배포받은 직원이 다른 직원에게 내부망을 통해 전송하는 방식으로만 설치가능하다.
 - 인접한 사무실끼리만 'SafeGuard' 프로그램을 송수신할 수 있다.
 - 인접한 사무실이란, 앞, 뒤, 왼쪽, 오른쪽, 위, 아래 사무실을 의미하며, 대각선 방향에 있는 사무실은 인접한 사무실이 아니다.
 - 보안프로그램 송수신 및 설치에는 12시간이 걸린다.
 - 초기에 safeguard가 설치된 사무실이더라도, 해당 사무실을 소유한 직원이 없을 경우에는, 프로그램을 다른 사무실로 전송하는 것은 불가능하다.
- (주)컴기전자에서 'SafeGuard'가 전 직원의 내부망PC에 설치완료될 수 있는지의 여부를 구하는 코드를 작성하라.
 - 모든 직원의 PC에 대해서 설치가 불가능할 경우에는, 설치불가능한 직원PC의 수를 프린트한다.
 - 모든 직원의 PC에 대해서 설치가 가능한 경우에는, 설치를 위해 소요되는 최소일수를 프린트한다.

Quiz

Q2.1 initial_stealthguard 함수를 완성하시오.

Hint. 3차원 매트릭스에서 행렬요소에 접근할 때는 [높이], [세로], [가로]좌표 순서로 접근하는 것이 일반적이다.

Q2.2 bfs_or_dfs 함수를 완성하시오.

Hint1. bfs_or_dfs()함수는 install_stealthguard(self) 함수 내부에서 정의되어 호출되는 함수이다.

Hint2. BFS, DFS 중 하나만 골라서 구현하면 된다.

Q2.3 install_stealthguard 함수를 완성하시오.

유의. print문은 변경하지 않는다.

Implementation:

```
M, N, H = 3, 3, 3
employees = [(1, 1, 1), (0, 0, 0), (1, 0, 2), (1, 1, 2), (0, 0, 2)]

office1 = Office(M, N, H, employees)
office1.initial_stealthguard([(1, 1, 1), (0, 0, 0), (1, 0, 2), (0, 0, 2)])
office1.install_stealthguard()
```

Python

1일 후면, 모든 사무실에 SafeGuard 프로그램이 설치됩니다.

```
M, N, H = 5, 3, 3
employees = [(0, 0, 0), (1, 0, 0), (2, 0, 0), (3, 0, 0)]

office2 = Office(M, N, H, employees)
office2.initial_stealthguard([(0, 1, 0), (1, 1, 0), (2, 1, 0), (3, 1, 0)])
office2.install_stealthguard()
```

Python

직원 4명의 PC에 SafeGuard 프로그램을 설치할 수 없습니다.

```
M, N, H = 5, 6, 3
employees = [(0,0,0),(0,1,0),(0,2,0),(0,4,0),(0,5,0),(1,0,0),(1,4,0),(1,5,0),
(2,0,0),(2,1,0),(2,3,0),(2,4,0),(3,1,0),(3,2,0),(3,3,0),(3,5,0),
(4,3,0),(4,4,0),(4,5,0),(0,0,1),(0,1,1),(0,2,1),(0,3,1),(1,1,1),
(1,3,1),(1,4,1),(3,2,1),(3,3,1),(3,4,1),(4,3,1),(4,4,1),(4,5,1),
(0,0,2),(0,3,2),(0,4,2),(1,0,2),(1,1,2),(1,4,2),(2,1,2),(2,2,2),
(2,3,2),(3,1,2),(3,4,2),(3,5,2),(4,0,2),(4,1,2),(4,2,2),(4,3,2)]

office5 = Office(M, N, H, employees)
office5.initial_stealthguard([(0,0,0),(1,1,1),(2,4,1),(4,0,2)])
office5.install_stealthguard()
```

Python

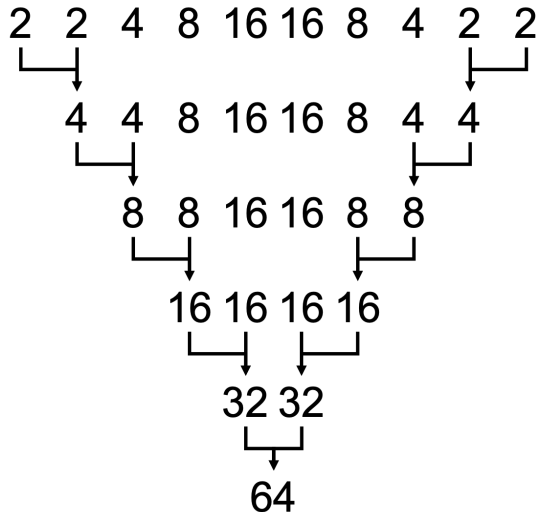
4일 후면, 모든 사무실에 SafeGuard 프로그램이 설치됩니다.

3. Merge Game [40pts]

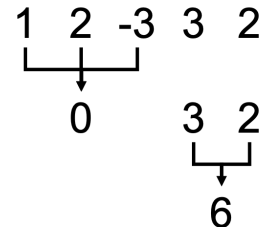
아래 지시문에 따라 각 게임의 결과를 출력하는 코드를 완성하시오.

Instruction:

Merge Game



Zero Sum Game



1. Merge Game은 연속된 두 숫자가 같은 경우 합쳐주었을 때 64라는 값이 하나라도 나오게 되면 이기는 게임으로 숫자는 2의 배수로만 주어지며 64를 완성하면 True를 아니면 False를 출력한다. (64를 초과하는 Test Case는 주어지지 않는다.)
2. Zero Sum Game은 연속된 숫자들의 합이 0이 되는 경우 배열에서 제외하고 남은 배열의 숫자들을 모두 곱한 값을 점수로 출력한다. (위 Case에서 1, 2, -3이 -3, 3보다 순서가 앞에 있기 때문에 1, 2, -3이 배열에서 제외된다.)
3. Zero Sum Game의 경우 Time Complexity $O(N)$ 으로 구현해야 한다. (그 이상 걸리는 경우 감점)
4. ##### 바깥 영역에 해당하는 코드는 수정이 불가능하다.

Quiz

Q3.1 MakeNode 함수를 완성하시오.

Hint. 주어진 ListNode class는 Single Linked List이므로 이전 Node에 접근이 어렵다.

Q3.2 MergeGame 함수를 완성하시오.

Hint. 연속으로 같은 값이 나오면 값을 합치고 연결도 새로 해주어야 한다. 다 합치고 나서 값이 64인 Node가 있는지 찾는다.

Q3.2 ZeroSumGame 함수를 완성하시오.

Hint. Hash Key에 누적합을 Value에 Node를 저장한다. 이후 누적합이 0이 되는 경우 연결을 새로 해준다.

Implementation:

```
arr = [2, 2, 4, 8, 16, 16, 8, 4, 2, 2]
MergeGame(arr)
```

✓ 0.0s

True

```
arr = [2, 2, 4, 8, 4, 4, 4, 16, 8, 4, 2, 2, 4]
MergeGame(arr)
```

✓ 0.0s

False

```
arr = [1, 2, -3, 3, 2]
ZeroSumGame(arr)
```

✓ 0.0s

6

```
arr = [1, 2, 3, -3, -2]
ZeroSumGame(arr)
```

✓ 0.0s

1

Cautions

1. 코드 채점 시에는, 예시로 주어진 **test case** 외에 추가적인 **test case**들을 활용하여 채점할 예정으로 **edge case**들에 대한 검증을 꼼꼼히 하고 제출하기 바랍니다.
2. 작성한 **ipynb** 파일을 **Assignment1_학생이름_학번.zip**로 압축하여 제출하기 바랍니다. (아래 예시 참고)

› Assignment1_이순신_2023-12345.zip

이름

Q1.ipynb

Q2.ipynb

Q3.ipynb

3. 제출 양식 미 준수 시 채점 간 오류가 발생할 수 있습니다. 제출 양식 미 준수로 인한 오류 교정은 없습니다. 수강생 분들은 제출 양식을 반드시 확인 바랍니다.