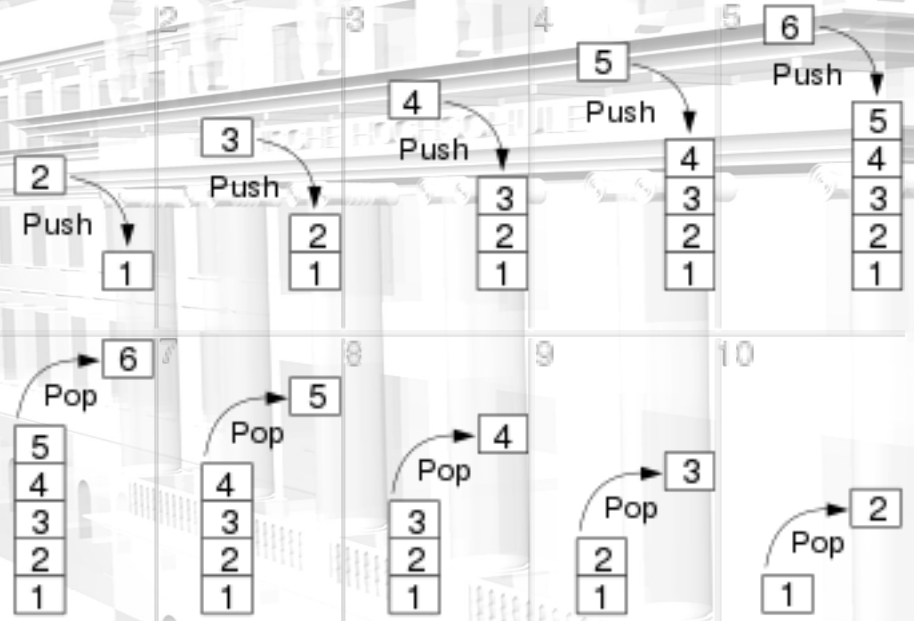
Pervasive Elastic MetaLearning Laboratory
Department of Computer Engineering
Hongik University

HW6 실습

Stack & Maze

Stack



- Linear Data Structure
- Last-In-First-Out: LIFO
- Push, Pop

https://en.wikipedia.org/wiki/Stack_(abstract_data_type)

Maze

entrance

0	1	0	0	0	1	1	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	0	0	1	1	1
0	1	1	0	0	0	0	1	1	1	1	0	0	1	1
1	1	0	1	1	1	1	0	1	1	0	1	1	0	0
1	1	0	1	0	0	1	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	0	1	0	0	1	0	1
0	0	1	1	0	1	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	0	1	1	1	1	1	1	1	1
0	0	1	1	0	1	1	0	1	1	1	1	1	0	1
1	1	0	0	0	1	1	0	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	1	1	1	1	0
0	1	0	0	1	1	1	1	1	0	1	1	1	1	0

exit

Figure 3.11: An example maze (can you find a path?)

Maze

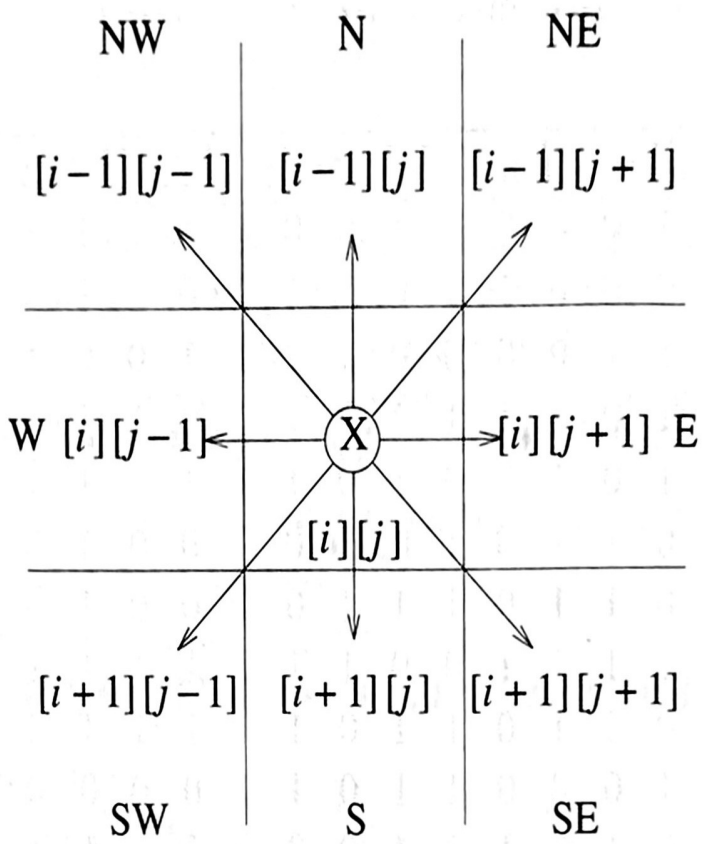


Figure 3.12: Allowable moves

q	$move[q].a$	$move[q].b$
N	−1	0
NE	−1	1
E	0	1
SE	1	1
S	1	0
SW	1	−1
W	0	−1
NW	−1	−1

Figure 3.13: Table of moves

Maze

entrance

0	1	0	0	0	1	1	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	0	0	1	1	1
0	1	1	0	0	0	0	1	1	1	1	0	0	1	1
1	1	0	1	1	1	1	0	1	1	0	1	1	0	0
1	1	0	1	0	0	1	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	0	1	0	0	1	0	1
0	0	1	1	0	1	1	1	0	1	0	0	1	0	1
0	1	1	1	1	0	0	1	1	1	1	1	1	1	1
0	0	1	1	0	1	1	0	1	1	1	1	1	0	1
1	1	0	0	0	1	1	0	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	1	1	1	1	0
0	1	0	0	1	1	1	1	1	0	1	1	1	1	0

exit

Figure 3.11: An example maze (can you find a path?)

```
void Path(const int m, const int p)
{
    mark[1][1] = 1;
    stack<Items> stack; // C++ STD stack을 이용하기
    Items temp(1, 1, E);
    stack.push(temp);
    //코드 입력
    while (!stack.empty()) {
        temp = stack.top(); stack.pop(); // unstack
        int i = temp.x; int j = temp.y; int d = temp.dir;
        while (d < 8) // move forward
        {
            int g = i + movea[d].a; int h = j + movea[d].b;
            if ((g == m) && (h == p)) { // reached exit
                cout << stack; // print reverse order of stack pops.
                // Now let's print the last two squares on the path
                temp.x = i; temp.y = j;
                cout << " -> " << temp;
                temp.x = m; temp.y = p;
                cout << " -> " << temp << endl;
                return;
            }
            if ((!maze[g][h]) && (!mark[g][h])) { // new position
                mark[g][h] = 1; // 방문한 적이 있다고 표시
                temp.x = i; temp.y = j; temp.dir = d + 1; // 현 위치와 실패 시
                    다음에 시도할 방향 저장
                stack.push(temp); // stack it
                i = g; j = h; d = N; // N 방향부터 (시계방향으로 ) 시도하자
            }
            else d++;
        } // end of while (d < 8)
    } // end of while (!stack.empty())
    cout << "No path in maze." << endl;
}
```

실습 설명

- 기술서에 적혀있는 대로 maze.cpp의 비어있는 부분을 작성할 것
- 더 생각해보고 싶은 학생은 ShortestPath() 함수 구현해보기
- 컴파일 방법: make hw6, 또는 Visual Studio 기능 사용

질문

- pemds81718@gmail.com
- 간단한 구글링으로 알 수 있는 내용은 답변하지 않습니다.
- 실습 내용 진행 중 궁금한 사항이 있을 경우 편하게 질문주세요.