

4주차: 스택(Stack)

1. 개념

▼ LIFO(Last In First Out)

- 나중에 입력한 데이터가 먼저 꺼내지는 자료구조
(먼저 입력한 데이터를 제일 나중에 꺼낼 수 있는 자료구조)
- 최근 입력 데이터를 가장 먼저 꺼낼 수 있는 자료구조
- 함수 호출, 페이지 탐색, 괄호 짝 맞추기 등에 활용
- DFS(Depth First Search, 깊이우선탐색), Back Tracking에서 활용
- 연산: PUSH(삽입), POP(꺼냄)

2. 정의

▼ 스택 라이브러리

- 파이썬은 스택 라이브러리 제공 안하여 대안 필요
 - append(), push() → stack() 대안 (한쪽으로만 데이터 삽입, 삭제)
- deque(): 덱 → 양쪽에서 데이터 삽입/삭제 가능한 자료구조

▼ 스택 세부 동작

- PUSH
 1. push()
 2. isFull() → check !
 3. top 증가
 4. top이 가리키는 위치에 push한 숫자 추가
- POP
 1. pop()
 2. isEmpty() → check !
 3. top 감소
 4. top이 가리키는 위치 숫자 반환

▼ ADT(Abstract Data Type, 추상 자료형)

- ADT: 자료 설계도. Interface만 있고 실제 구현은 되지 않는 자료형
- 스택에 대한 ADT 작성 → 스택 정의
 - push(), pop(), isFull, isEmpty 연산 정의 필요
 - 최근 삽입한 데이터 저장소(변수) top 필요
 - 연산관련

```
# 연산
void push(ItemType item) #스택에 데이터 푸시

ItemType pop() #스택에 최근에 푸시한 데이터 팝. 그 데이터 반환

boolean isFull() #스택 데이터개수 maxsize면 True, 아니면 False

boolean isEmpty() #스택이 비었으면 True, 아니면 False

# 상태
int top #스택에 푸시한 데이터 위치 기록

ItemType data[maxsize] #스택 데이터 관리 배열. 최대 maxsize
```

▼ 스택 ADT 구현

```
stack = []
max_size = 10

def isFull(stack):
    return len(stack) == max_size

def isEmpty(stack):
    return len(stack) == 0

def push(stack, item):
    if isFull(stack):
        print("stack is full")
    else:
        stack.append(item)
        print("data is added")
```

```
def pop(stack):
    if isEmpty(stack):
        print("stack is empty")
        return None
    else:
        return stack.pop()
```

3. 기본 예제

▼ 괄호 짝 맞추기

```
from collections import deque

fname = '/content/drive/MyDrive/Colab Notebooks/Algorit
solution():
    with open(fname, 'r') as f:
        for _ in range(2): #test case = 2
            deq = deque()
            input_data = []
            input_data = list(map(str, f.readline().str
            for i in range(len(input_data)):
                if input_data[i] == '(':
                    deq.append('(')
                    #print('left',deq)
                else:
                    deq.pop()
                    print('right1',deq)
                    if '(' in deq and input_data[i+1] ==
                        #print('right2',deq)
                        #print('nonono')
                        break
            #print(bool(len(deq) == 0))
            return bool(len(deq) == 0)
```

▼ 10진수 to 2진수

```
solution():
    with open(fname, 'r') as f:
```

```

input_data = []
T = int(f.readline())
for tc in range(T):
    input_data = int(f.readline())
    deq = deque()
    while True:
        if input_data // 2 == 0:
            break
        else:
            deq.append(input_data%2)
            input_data = input_data//2
    deq.append(1)
    rst = []
    for _ in range(len(deq)):
        rst = deq.pop()
        print(rst, end='')
    print()

```

4. 모의 테스트

▼ 괄호 회전

```

'''
올바른 괄호 문자열: (A), [A], {A}, AB, {}([])
문자열s (괄호) -> 왼쪽으로 x(s의 길이)칸 만큼 회전 -> 올바른 괄호

[input]
[](){}
}]()[{
]()[]
}}}

[output]
3
2
0
0
'''
from collections import deque

```

```

fname = '/content/drive/MyDrive/Colab Notebooks/Algorit
with open(fname, 'r') as f:
    T = int(f.readline())
    for tc in range(1, T+1):
        input_data = f.readline()
        deq = deque(input_data.strip())
        cnt = 0
        for idx in range(len(deq)):
            if deq[0] in ['(', '{'] and deq[-1] in [
                # print(idx, deq)
                cnt += 1
                temp = deq.popleft()
                deq.append(temp)
        print(cnt)

```

▼ 짝지어 제거

```

'''
알파벳 소문자
같은 알파벳 2개 붙어있는 짝 찾음 -> 제거 -> 앞뒤로 문자열을 이어 붙임
문자열 s -> 짝지어 제거 수행 여부 반환(1 or 0) 함수 출력
baabaa -> bbbaa -> aa (순서대로 제거. 제거 성공하므로 1 반환)

[input]
baabaa
cdcd

[output]
1
0
'''
from collections import deque

fname = '/content/drive/MyDrive/Colab Notebooks/Algorit
with open(fname, 'r') as f:
    T = int(f.readline())
    for tc in range(1, T+1):

```

```

input_data = f.readline().strip()
deq = deque()
for i in range(len(input_data)):
    if input_data[i] != input_data[i-1]:
        deq.append(input_data[i])
    else:
        deq.pop()
        if input_data[i] == input_data[i-1]:
            deq.pop()
        # print('deq:', deq)
print(bool(len(deq)==0))

```

▼ 주식 가격

```

#include <string>
#include <vector>
#include <stack>

using namespace std;

vector<int> solution(vector<int> prices) {
    vector<int> answer(prices.size());
    stack<int> s;

    int priceNum = prices.size();

    for(int i=0; i<priceNum; i++){
        while(!s.empty() && prices[s.top()] > prices[i]){
            answer[s.top()] = i - s.top();
            s.pop();
        }
        s.push(i);
    }
    while(!s.empty()){
        answer[s.top()] = priceNum - s.top() - 1;
        s.pop();
    }
    return answer;
}

```

```

}

/*Test*/
#include <iterator>
#include <iostream>
void print(vector<int> vec)
{
    copy(vec.begin(), vec.end(), std::ostream_iterator<
    cout << endl;
}

int main()
{
    print(solution({1, 2, 3, 2, 3})); // 4 3 1 1 0
    return 0;
}

```

▼ 크레인 인형 뽑기 게임

```

#include <stack>
#include <vector>

using namespace std;

int solution(vector<vector<int>> board, vector<int> move) {
    stack<int> lanes[board[0].size()];
    for(int i = board.size()-1 ; i >= 0; --i) {
        for(int j = 0; j<board[0].size(); ++j) {
            if(board[i][j]) {
                lanes[j].push(board[i][j]);
            }
        }
    }

    stack<int> bucket;
    int answer = 0;
}

```

```

for(int m : moves) {
    if(lanes[m-1].size()){
        int doll = lanes[m-1].top();
        lanes[m-1].pop();
        if (bucket.size() && bucket.top() == doll) {
            bucket.pop();
            answer += 2;
        } else {
            bucket.push(doll);
        }
    }
}

return answer;
}

/*Test*/
#include <iostream>

int main()
{

    cout << solution( { {0, 0, 0, 0, 0}, {0, 0, 1, 0, 3}
    return 0;
}

```

▼ 표 편집

```

#include <string>
#include <vector>
#include <stack>

using namespace std;

string solution(int n, int k, vector<string> cmd) {
    stack<int> deleted;
    vector<int> up;
}

```



```

vector<int> down;

for (int i = 0; i < n + 2; i++) {
    up.push_back(i - 1);
    down.push_back(i + 1);
}
k++;

for (int i = 0; i < cmd.size(); i++) {
    if (cmd[i][0] == 'C') {
        deleted.push(k);
        down[up[k]] = down[k];
        up[down[k]] = up[k];

        if (down[k] == n + 1) k = up[k];
        else k = down[k];
    }

    else if (cmd[i][0] == 'Z') {
        int r = deleted.top();
        down[up[r]] = r;
        up[down[r]] = r;
        deleted.pop();
    }

    else {
        int sz = stoi(cmd[i].substr(2));

        if (cmd[i][0] == 'U') {
            for (int j = 0; j < sz; j++) {
                k = up[k];
            }
        }

        else if (cmd[i][0] == 'D') {
            for (int j = 0; j < sz; j++) {
                k = down[k];
            }
        }
    }
}

```

```

        }
    }

}

string answer;

answer.append(n, '0');
while (!deleted.empty()) {
    answer[deleted.top() - 1] = 'X';
    deleted.pop();
}

return answer;
}

/*Test*/
#include <iostream>

int main()
{
    cout << solution(8, 2, {"D 2", "C", "U 3", "C", "D .
    cout << solution(8, 2, {"D 2", "C", "U 3", "C", "D .
    return 0;
}

```