

# 알고리즘 스터디 5주차(2493, 9012, 1874번)

2493번\_탑 (<https://www.acmicpc.net/problem/2493>)

## 문제 접근 방식

- 스택특집이라는 힌트가 있으므로, 스택으로 접근합니다.
- 스택 이용합니다. 탑을 등장순으로 스택에 넣어두면, 스택의 최상단은 왼쪽 탑이 됩니다. 그러면 이후 스택의 위에서 부터 해당 값 보다 큰 값을 찾습니다. 큰 값이 나온 인덱스를 정답용 스택에 push합니다. 위 작업의 반복이 끝나, 모든 탑들의 수신탑번호를 찾게되면, stack를 pop하는 순서대로 출력하면 됩니다.

## 중요함수

```
void find() {
    for (int i = 0; i < N; i++) {
        int hight = tower.top();
        tower.pop();
        int index = tower.size();

        for (; tower.empty() == 0; index--) {
            int tempTower = tower.top();
            tower.pop();
            temp.push(tempTower);
            if (tempTower > hight) {
                answer.push(index);
                break;
            }
        }

        for (; temp.empty() == 0;) {
            tower.push(temp.top());
            temp.pop();
        }

        if (N != answer.size() + tower.size()) answer.push(0);
    }
}
```

-> 시간초과가 났다. 여기서 더 줄여보자. (현재  $O(n^2)$ )

-> tower의 자료구조를 vector로 바꾸고, push\_back(), pop\_back()를 이용해도 안됨ㅋ

필살기 scanf(), printf()를 사용 - 시간초과ㅋㅋㅋㅋ

메모리 제한이 매우 넉넉하므로 DP 를 사용해 보자

DP 생각이 어렵다. 나중에 다시도전하겠다 시팔.

## 9012번\_괄호 (<https://www.acmicpc.net/problem/9012>)

### 문제 접근방식

- (와 )의 개수가 일치하고, 시작은 (, 끝은 )로 되는 경우를 찾으면 됩니다.
- int형 변수 LPS(left PS)를 이용합니다. (가 발견되면 LPS를 +1, )가 발견되면 LPS 감소시킵니다. LPS가 RPS보다 작은 경우는 올바른 괄호문자열 이 아니므로, 위 방법으로 문자열의 앞에서부터 읽어 나갑니다.

### 중요코드

```
string str;

for (int test_case = 0; test_case < N; test_case++) {
    LPS = 0;
    cin >> str;

    for (int i = 0; i < str.size() && 0 <= LPS; i++) {
        if (str[i] == '(') LPS++;
        else LPS--;
    }
    if (LPS == 0) printf("YES\n");
    else printf("NO\n");
}
```

```
#include <iostream>
#include <string>
int main(){int N,L,i;std::cin>>N;std::string s;for(;0<N;N--){L=0;std::cin>>s;for(i=0;i<s.size()&&L>-1;i++){if(s[i]=='(')L++;else L--;}if(L==0)puts("YES\n");else printf("NO\n");}}
```

쑈코딩일등하는새끼들은 뭐하는새끼까

## 1874번\_스택 수열(<https://www.acmicpc.net/problem/1874>)

### 문제접근 방법

- 문제에서 '스택에 push하는 순서는 반드시 오름차순을 지키도록 한다고 하자' 부분이 이해가 안되므로, 예제를 보며 하나씩 해봅시다.

-입력 8,[4,3,6,5,8,7,5,2,1]에서 스택과 수열

	스택	수열
+	1	
+	{1, 2}	
+	{1, 2, 3}	
+	{1, 2, 3, 4}	
-	{1, 2, 3}	4
-	{1, 2}	4 3
+	{1, 2, 5}	4 3
+	{1, 2, 5, 6}	4 3
-	{1, 2, 5}	4 3 6
+	{1, 2, 5, 7}	4 3 6
+	{1, 2, 5, 7, 8}	4 3 6...
....		

- stack 두개를 이용한다. input값을 저장할 스택 input과 문제를 해결하는데 사용할 스택 temp을 사용한다

- 주어진 수열을 stack에 넣어서 stack의 top부터 일치시켜나간다.

### 중요코드

```
//temp_stack.top()와 input[i]의 관계는 1. 같은경우, 2. 작은경우, 3. 큰경우.
//4. 없는경우 - temp_stack에 0을 미리 푸쉬한상태로 시작하기에 존재하지 않음
bool makeSequence(int index, int num) {
    if (temp_stack.top() == input[index]) {
        temp_stack.pop();
        answer.push_back('-');
        if (index == N) return true;
        makeSequence(index + 1, num);
    }
    else if (temp_stack.top() < input[index]) {
        temp_stack.push(num);
        answer.push_back('+');
        makeSequence(index, num + 1);
    }
    else if (temp_stack.top() > input[index]) {
        return false;
    }
}
```

```
}
```

시간초과 났다 이것도... DP써보자? → 겹치는 부분이 없어서 사용못한다

Printf랑 scanf하니까 바로통과한다. 왜그럴까?

std::ios\_base::sync\_with\_stdio(false);는 효과가 있는건지 없는건지 매번사용해야하는건지 알쏭달쏭하다. Cin과 cout가 문제가 안되도록 생각해보자

아니면 중도여서 와이파이가 느린것도 채점속도에 영향을 미치나?

getchar, putchar을 써보면 더빨라지겠지?

**<<endl을 쓰면 매우 느려짐을 알게되었다!! endl보단 'wn'**

endl을 쓸때는 시간초과였던게 wn을 사용해보니 시간초과 x