

# 스택 : 짝지어 제거하기

## 문제 설명

짝지어 제거하기는, 알파벳 소문자로 이루어진 문자열을 가지고 시작합니다. 먼저

예를 들어, 문자열  $S = \text{baabaa}$  라면

$b \text{ aa baa} \rightarrow bb \text{ aa} \rightarrow aa \rightarrow$

의 순서로 문자열을 모두 제거할 수 있으므로 1을 반환합니다.

## 제한사항

문자열의 길이 : 1,000,000이하의 자연수

문자열은 모두 소문자로 이루어져 있습니다.

입출력 예

| s      | result |
|--------|--------|
| baabaa | 1      |
| cdcd   | 0      |

## 1. 오답

→ 문자열 전체 순회해서  $i$ 번째와  $i+1$ 번째 문자가 같으면 제거하고, 다시 처음부터 순회하도록 구현

→ 효율성 낮음, 시간복잡도  $O(n^2)$

```
#include <iostream>
#include<string>
using namespace std;

int solution(string s)
{
    int answer = 1;
    int i=0;
```

```

while(i<s.size()-2){
    if(s.empty()){
        answer=1;
        break;
    }

    if(s[i]==s[i+1]){
        s.erase(i,2);
        i=0;
        continue;
    }

    else{
        i++;
    }
}

if(!s.empty()){answer=0;}

return answer;
}

```

## 2. 정답

→ 스택 활용하기

→ 문자열 순회하는데

→스택이 비어있거나 / 스택 맨 위의 값과 문자가 다르면 push

→문자가 같으면 pop

→최종적으로 스택이 비어있으면 1을 반환 , 아니면 0 반환

```

#include <iostream>
#include<string>
#include<stack>
using namespace std;

```

```
int solution(string s)
{
    int answer = 1;
    if(s.size()%2 != 0){
        return 0;
    }
    stack<char> stk;
    for(int i=0; i<s.size();i++){
        if(stk.empty()||stk.top()!=s[i]){
            stk.push(s[i]);
        }
        else{
            stk.pop();
        }
    }
    if(!stk.empty()){
        return 0;
    }
    return answer;
}
```