

# 괄호 ( )

## *BaekJun 9012*



# 목차

---

01

문제 분석

---

02

시간복잡도  
분석

---

03

문제 풀이

---

04

후기

# 1. 문제 분석

## 문제

괄호 문자열(Parenthesis String, PS)은 두 개의 괄호 기호인 '(' 와 ')' 만으로 구성되어 있는 문자열이다. 그 중에서 괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열 (Valid PS, VPS)이라고 부른다. 한 쌍의 괄호 기호로 된 "( )" 문자열은 기본 VPS 이라고 부른다. 만일 x 가 VPS 라면 이것을 하나의 괄호에 넣은 새로운 문자열 "(x)"도 VPS 가 된다. 그리고 두 VPS x 와 y를 접합(concatenation)시킨 새로운 문자열 xy도 VPS 가 된다. 예를 들어 "(()())"와 "((()))" 는 VPS 이지만 "(()(", "(()())()", 그리고 "(" 는 모두 VPS 가 아닌 문자열이다.

여러분은 입력으로 주어진 괄호 문자열이 VPS 인지 아닌지를 판단해서 그 결과를 YES 와 NO 로 나타내어야 한다.

중요 포인트: *VPS*

# VPS란?

괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열

Example:

1. 입력: (((()))) -> 출력: Yes

2. 입력: (()()) -> 출력: No

'('와 ')'의 갯수가 같으면 YES 출력, 같지 않으면 NO 출력

## 2. 시간복잡도 분석

이중반복문을 사용해 문제 풀이를 해야하므로  $n \times n$ 만큼 돌아갔을 때 최대로 돌아가므로 시간복잡도는

$O(n^2)$

### 3. 문제 풀이

#### 풀이 방법:

stack 함수를 사용하여  
'('와 ')'의 갯수가 같으면, 즉 VPS가 맞다면 pop이 될 수  
있도록 이중반복문 안에 if문으로 VPS가 맞는지 아닌지  
판단하여서 Yes 또는 NO를 출력한다.

# CODE:

times 변수 선언



문자열 선언



times만큼 반복



VPS인지 판단하는 스택 선언



```
int times = 0;
```

```
int flag = 0;
```

```
string str;
```

```
cin >> times;
```

```
for (int i = 0; i < times; i++)
```

```
{
```

```
    stack<char> judgeVPS;
```

```
    cin >> str;
```

j 선언 및 입력값의 길이만큼 반복 →

str[j]가 '('이면 push →

str[j]가 ')' 였을 때, 스택이 비어  
있다면 오류가 나기 때문에  
flag를 1로 두고 break →

'('가 push 되어있다면 pop →

flag가 1이거나, 스택이 비어있지  
않으면 "NO"출력 →

VPS가 성립된다면 "YES" 출력 →

flag를 초기화시킨 뒤 다시 상위  
for문으로 이동 →

```
for (int j = 0; j < str.size(); j++)
{
    if (str[j] == '(') {
        judgeVPS.push(str[j]);
    }
    else if (str[j] == ')')
    {
        if (judgeVPS.empty())
        {
            flag = 1;
            break;
        }
        else
        {
            judgeVPS.pop();
        }
    }
}

if (!judgeVPS.empty() || flag == 1)
{
    cout << "NO" << endl;
}
else
{
    cout << "YES" << endl;
}

flag = 0;
```



## 4. 후기

**이번 기회로 알고리즘에 흥미를 갖게 되는 계기가  
되었고 민서 선배님과 친구들이 모르는 부분을  
친절히 알려주셔서 이해가 쉬웠던 것 같습니다.  
비록 저의 역량 부족으로 블로그 글을 참고했지만  
코드 하나를 꼭꼭 씹으며 성장할 수 있는 계기가 되었던 것 같습니다.**

# 감사합니다

드디어 끝났다 ㅏ

