

String (시험에 나오는거 위주로)

1. 개념

- 1) `charAt(i)` => 문자열의 위치
- 2) `charAt(i) - '0' => int값`
- 3) `toCharArray()` => `char[]` return
- 4) `IndexOf()`, `subString()`
- 5) Palindrome
ex) banana => anana

1. String

Problem

Input: 1->2->3,
8->8->8

Output: 1011

문자열처럼 생긴 음이 아닌 정수값 num1, num2
(non-negative integers num1 and num2 represented as string)
문자열로 num1, num2 의 합을 return

Note

- 1) num1 및 둘 다의 길이 num2는 <1000입니다.
- 2) 둘 다 num1 및 num2 숫자 만 포함합니다 0-9.
- 3) num1 및 둘 다 num2 선행 0을 포함하지 않습니다

문제 Format

```
class Solution {  
    public String solve(String num1, String num2) {  
    }  
}
```

Solution

String

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for ,While 문 돌리고 그 안에 알고리즘 넣기

I Can Image

- . 내가 생각한거 적고->프로그램화(한국말로 생각하고->Java)
결과를 해석하여 이미지화시킨다

1. 구현할 내용을 써보자~

String

`charAt()`, `substring()`

`int num = str.charAt(i) - '0';`

1

0

1

8

1

1

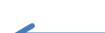
2

8

2

3

8



1

`int sum = n1 + n2 + carry;`

2

`carry = sum / 10;`

3

`sb.append(sum % 10);`

1

0

1

1

2. String

Problem

Input: String s = "in(f(lea)r)n";

Output: "in(f(lea)r)n"

Input: String s = "(a(b(c)d)";

Output: "(a(bc)d)"

최소 괄호의 수를 제거하여 '(' ')' 처럼 유효한 parentheses string를 리턴한다

1. 빈 문자열이거나 소문자 만 포함
2. (ab), ab(c)d

문제 Format

```
class Solution {  
    public String solve(String str) {  
    }  
}
```

String

toArray()

String s = "(a(b(c)d)";

(a (b (c) d)

1. 가로에 개수를 파악한다.
2. () 쌍을 이용하여 (의 개수 count =1 을 저장
3. 역순으로 빼낸다
4. 2번에서 구한 "(" 이면서 count =1인 경우 저장하지 않는다
5. 역순으로 저장된 값에 "(" 개가 제외된 상태이므로
다시 reverse해서 리턴한다.

3. String

Problem

Input: `String str[] = { "test" , "teacher" }`

Output: `"te"`

앞자리 te를 리턴한다

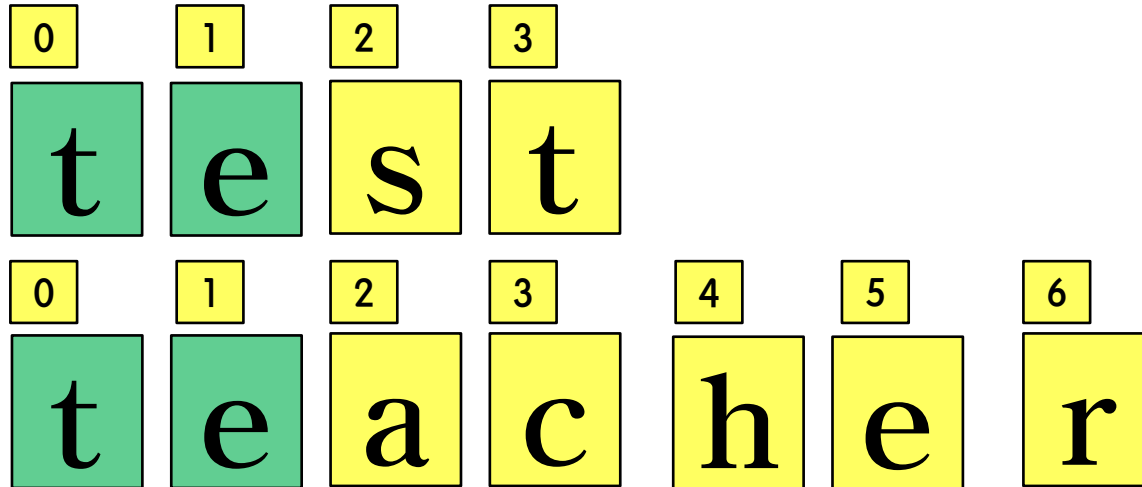
문제 Format

```
class Solution {  
    public String solve(String[] strs) {  
    }  
}
```

String

indexOf()

substring()



indexOf()

1. 문자의 위치를 리턴, 없으면 -1

substring()

1. `substring(0, 7)` = teacher

String

indexOf()

substring()

0 1 2 3 4 5 6
t e a c h e r

0 1 2 3
t e s t

0 1 2
t e s

0 1
t e

4. String

Longest palindrome substring

Problem

Input: s = "bananas"

Output: "anana"

Input: s = "babad"

Output: "bab"

return the longest palindromic substring in s

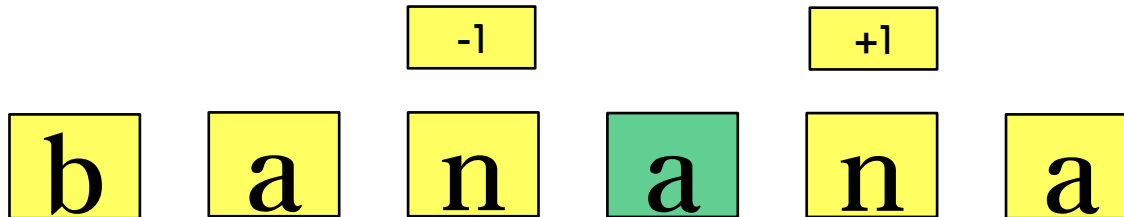
Note

문제 Format

```
class Solution {  
    public String solve(String s) {  
    }  
}
```

String

index



String

index



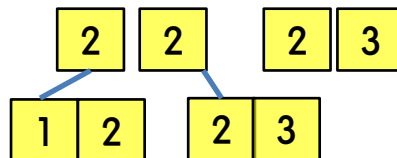
Start:0 end:1



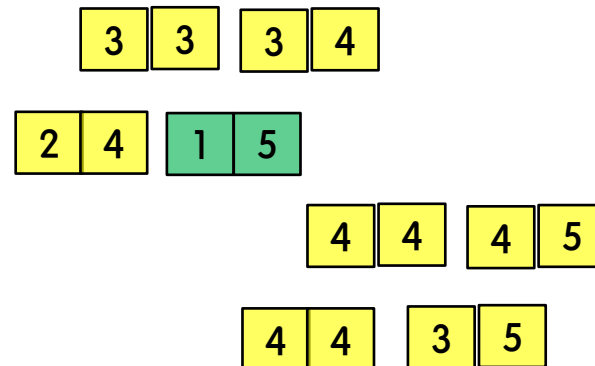
Start:0 end:1



Start:1 end:3



Start:1 end:5



String

index

0

b

1

a

2

n

3

a

4

n

5

6

a

b

a

n

a

n

a

