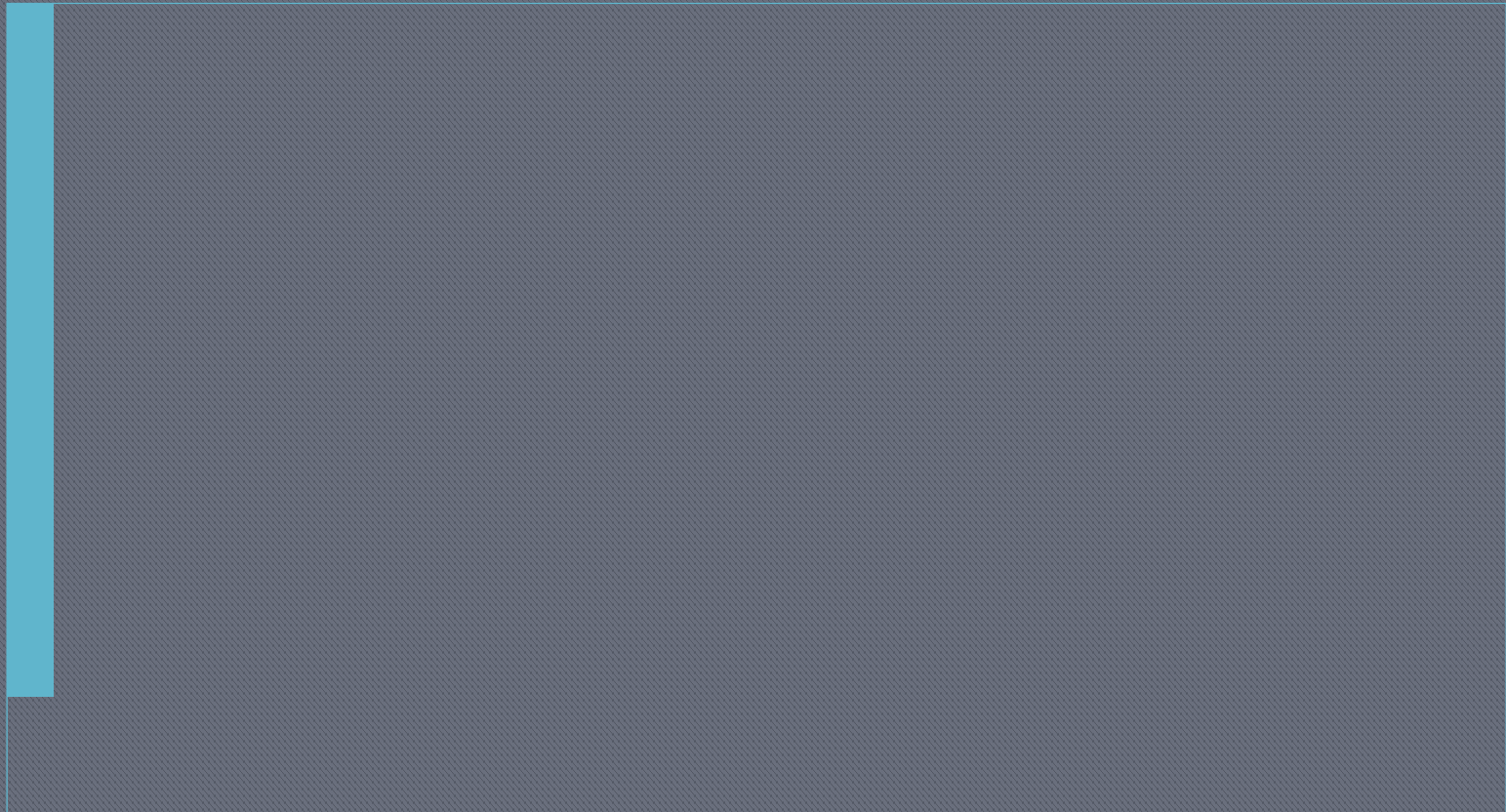




# 재귀함수의 흐름

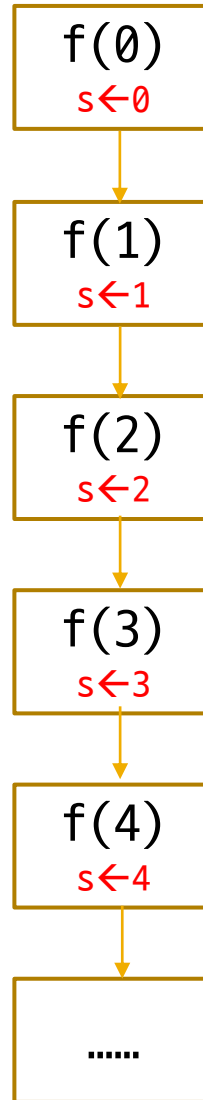


# f(0)이 호출된다면...

---

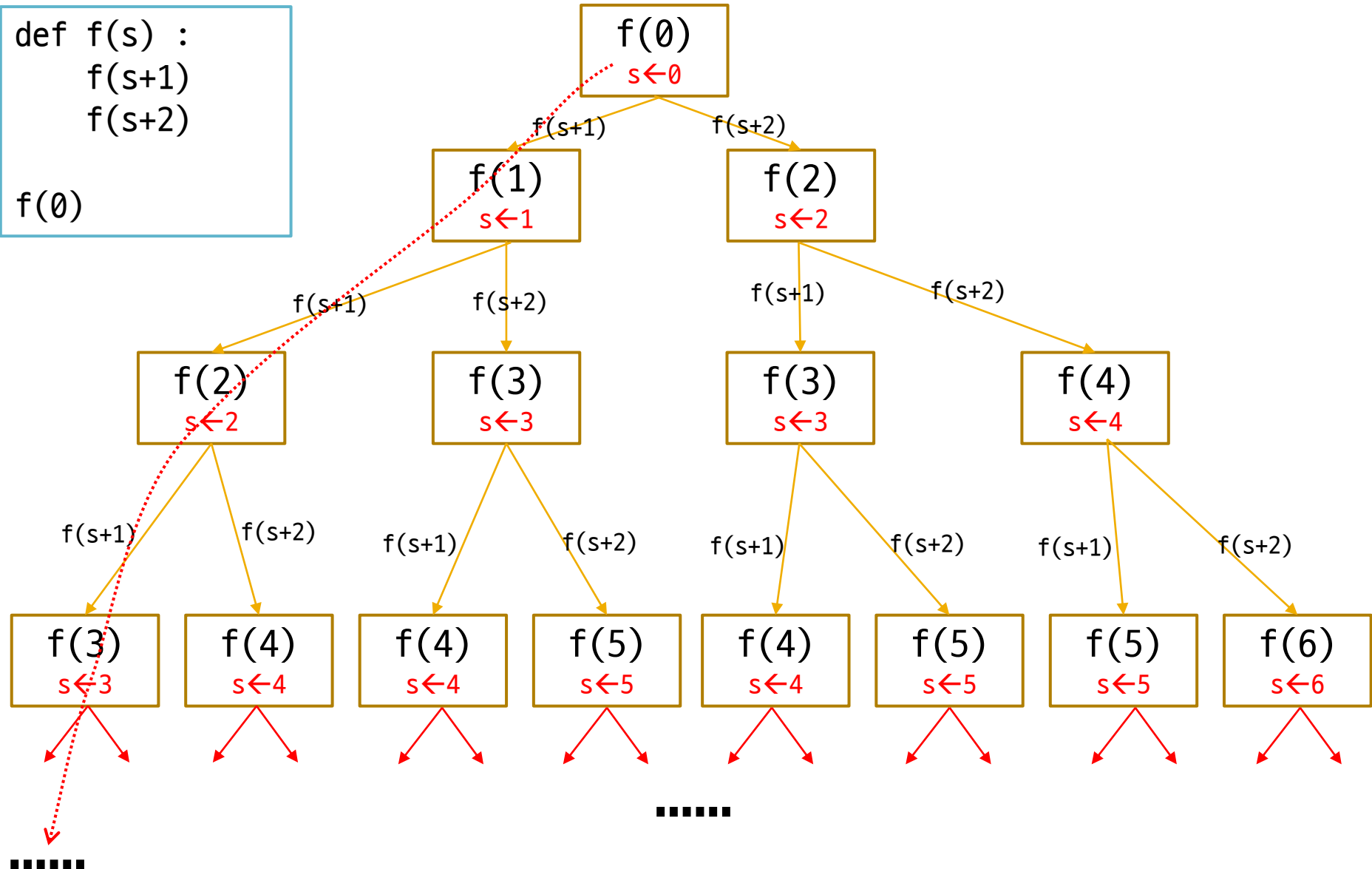
```
def f(s) :  
    f(s+1)
```

f(0)



# f(0)이 호출시 아래와 같은 호출구조를 가지려면???

```
def f(s) :  
    f(s+1)  
    f(s+2)  
  
f(0)
```



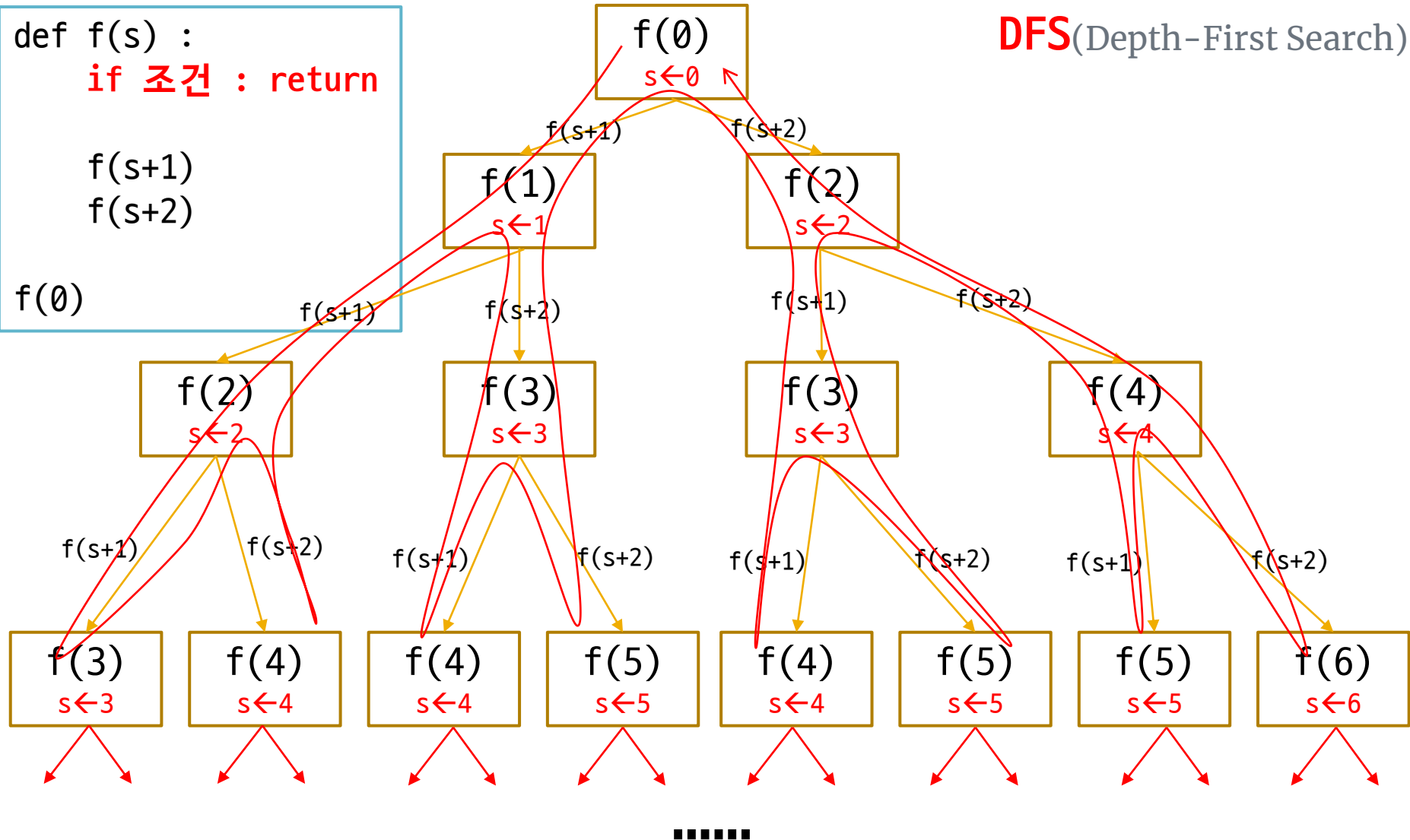
# f(0)이 호출시 아래와 같은 호출구조를 가지려면???

```
def f(s) :  
    if 조건 : return
```

```
    f(s+1)  
    f(s+2)
```

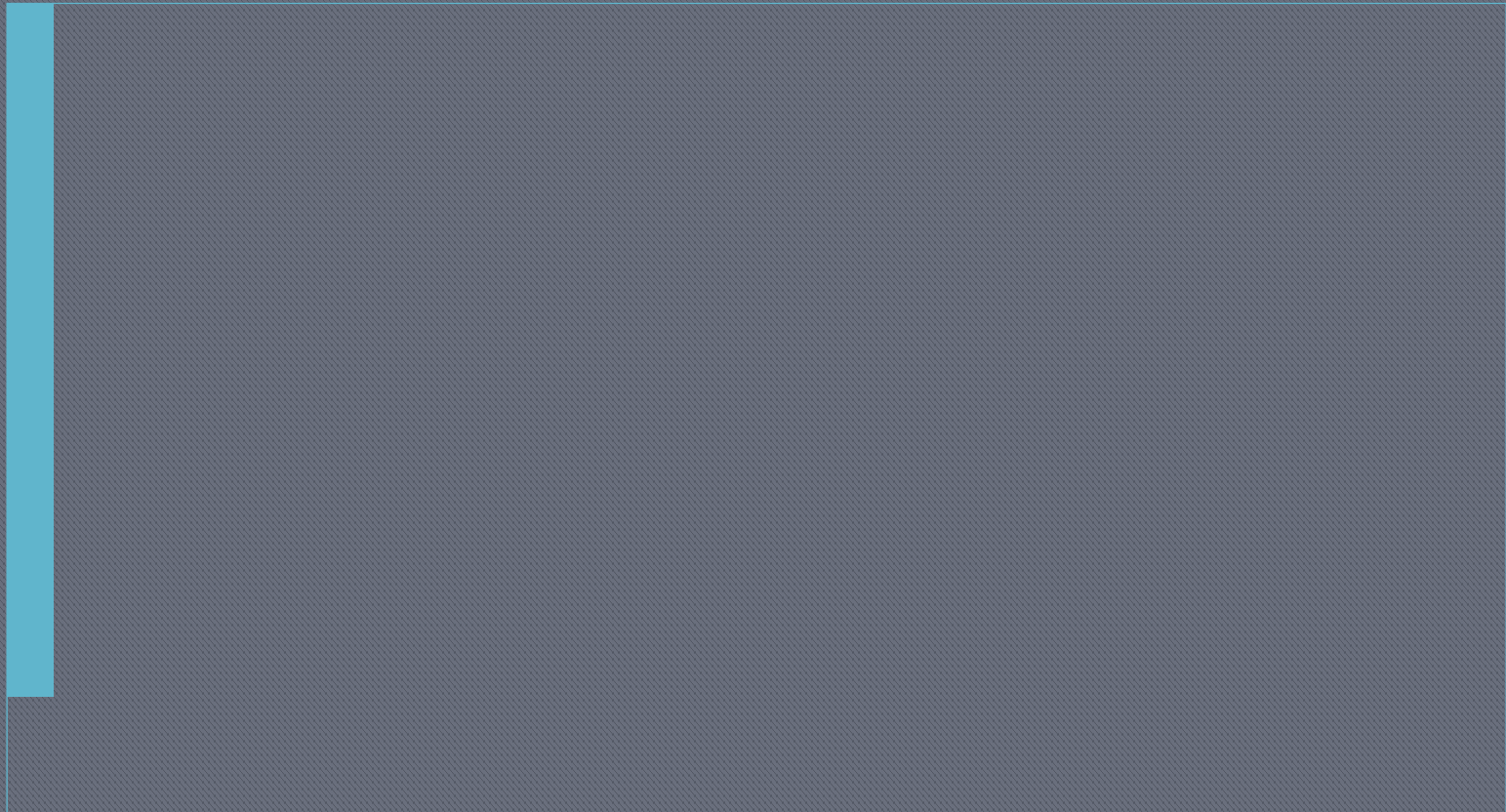
f(0)

**DFS**(Depth-First Search)





# 계단 오르기



# 문제 AD: [재귀-기초] 계단 오르기(모든 방법 출력) (python)

시간제한: 1.0sec

메모리제한: 128MiB

통과: 2

제출: 638

제출

제출화면

문제리스트

문제수정

TESTDATA

## 문제 설명

한 번에 계단을 1개 또는 2개 또는 3개를 뛰어 오를 수 있을 때,  
한 자연수  $n$ 을 입력받아 바닥(0번째 계단)에서  $n$ 번째 계단까지 도착할 수 있는 모든 방법들과 그 수를 출력하시오.

예를 들어,

$n$ 이 1이면 : 1번 계단에 도착하는 방법은 1가지(1칸)이다.

$n$ 이 2이면 : 2번 계단에 도착하는 방법은 2가지(1칸-1칸, 2칸)이다.

$n$ 이 3이면 : 3번 계단에 도착하는 방법은 4가지(1칸-1칸-1칸, 1칸-2칸, 2칸-1칸, 3칸)이다.

## 입력/출력 데이터 형식

### 입력 데이터 형식

자연수  $n$ 을 입력받는다. ( $1 \leq n \leq 20$ )

### 출력 데이터 형식

첫번째 줄에는 바닥(0번)에서  $n$ 번 계단까지 오르는 방법의 수를 출력한다.

두번째 줄에는  $n$ 번 계단에 도착할 수 있는 방법들을 1줄에 1가지씩 출력한다.

(출력할 때는, 사전순으로 출력한다. (작은수 --> 큰 수))

## 입력/출력 데이터 예시

>> 입력 데이터 예시1  복사

2

>> 입력 데이터 예시2  복사

3

>> 입력 데이터 예시3  복사

6

<< 출력 데이터 예시1  복사

1 1  
2  
2

<< 출력 데이터 예시2  복사

1 1 1  
1 2  
2 1  
3  
4

<< 출력 데이터 예시3  복사

1 1 1 1 1 1  
1 1 1 1 2  
1 1 1 2 1  
1 1 1 3  
1 1 2 1 1  
1 1 2 2  
1 1 3 1  
1 2 1 1 1  
1 2 1 2  
1 2 2 1

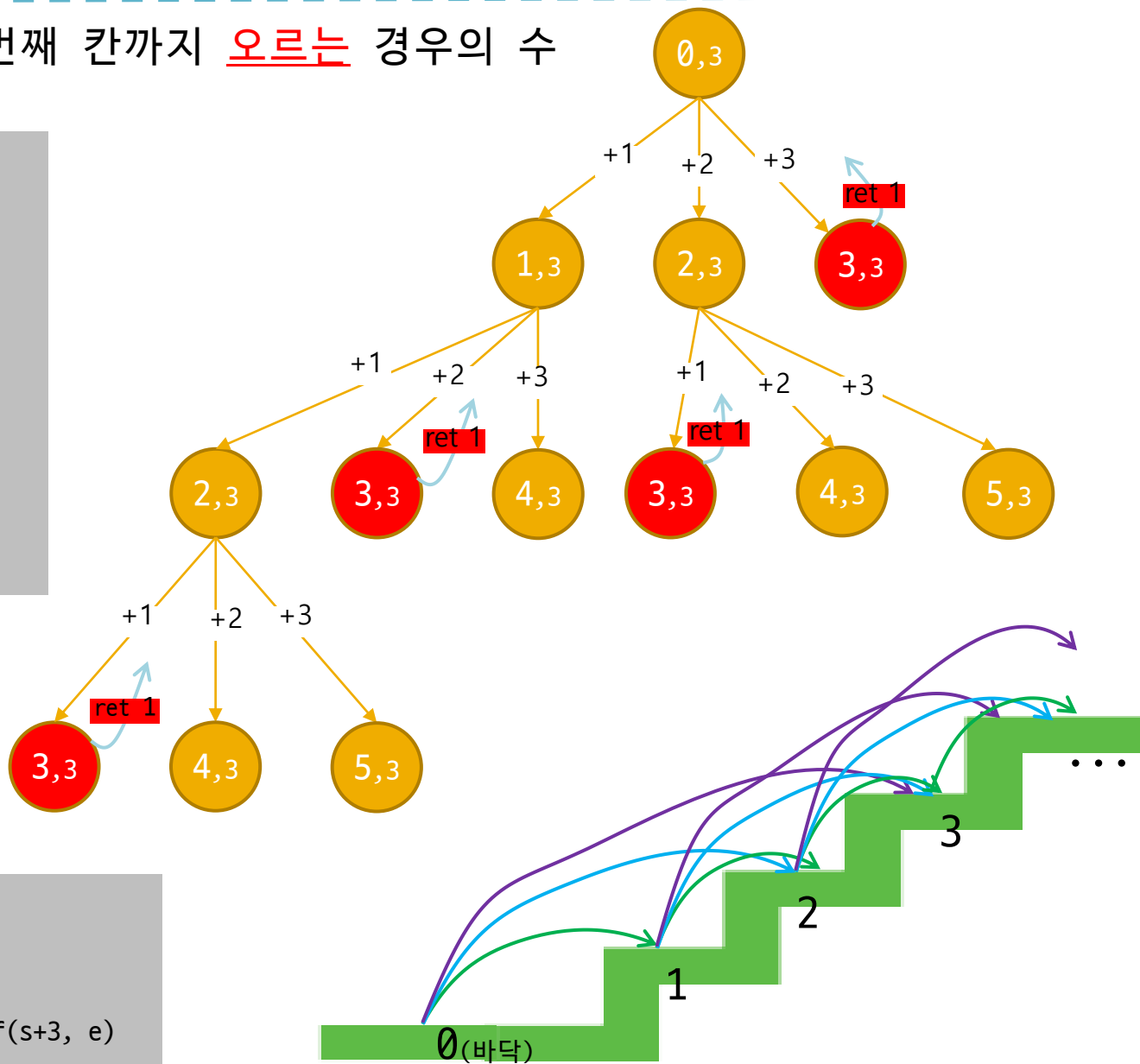
# 계단오르기 - 경우의 수 구하기

$f(0, 3)$  : 0번에서 3번째 칸까지 오르는 경우의 수

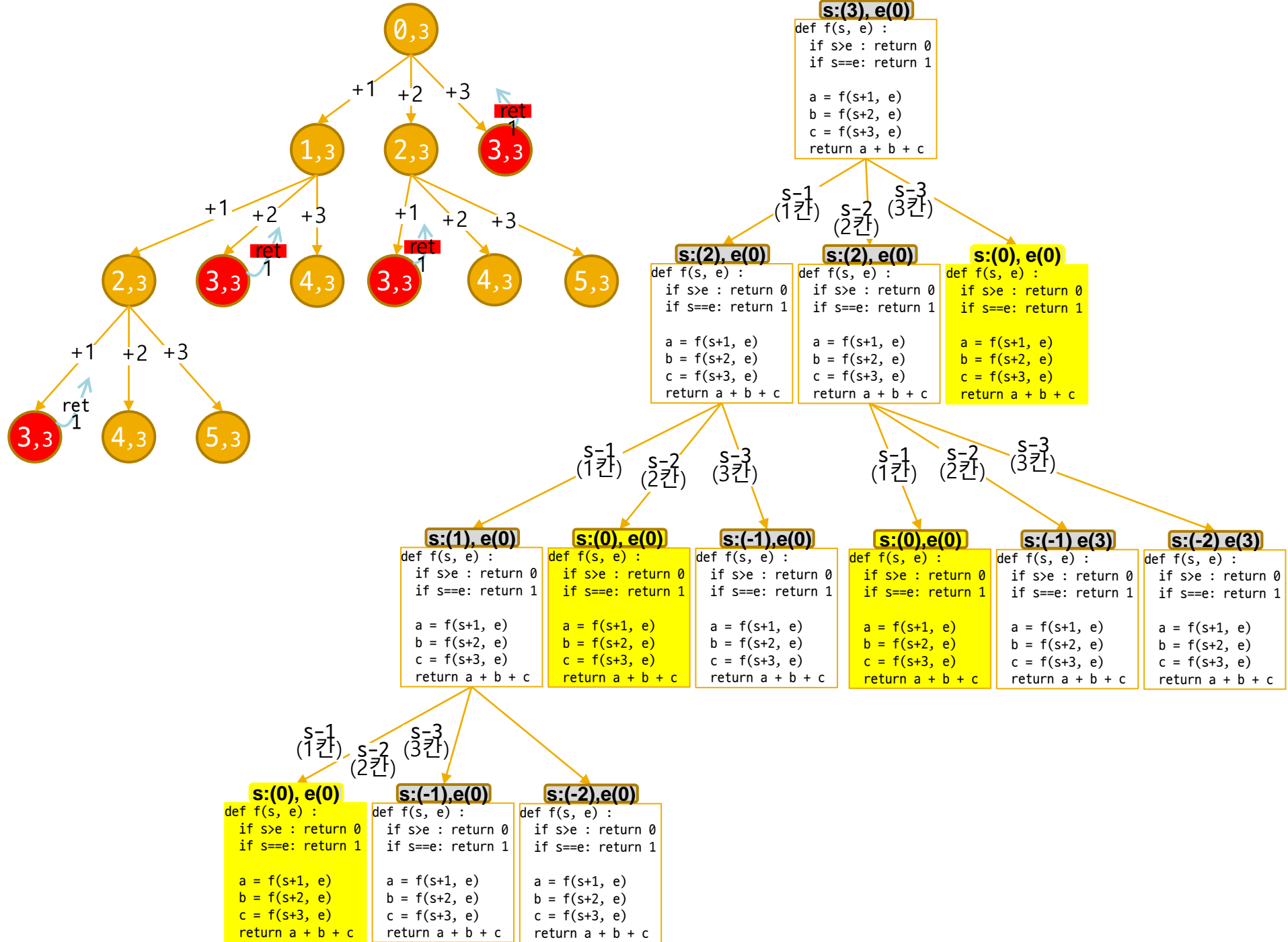
```
def f(s, e) :  
    if s > e : return 0  
    if s == e : return 1
```

```
    a = f(s+1, e)  
    b = f(s+2, e)  
    c = f(s+3, e)  
    return a + b + c
```

```
print(f(0, 3))
```







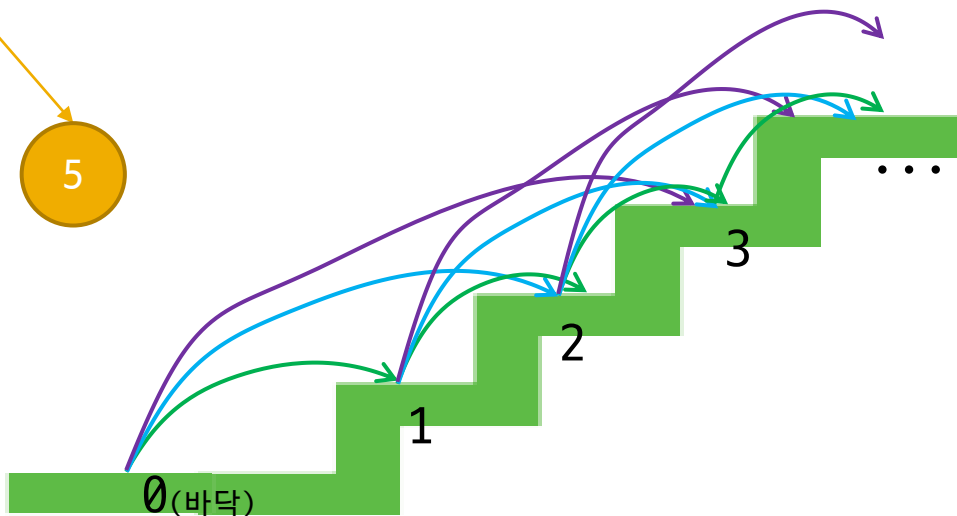
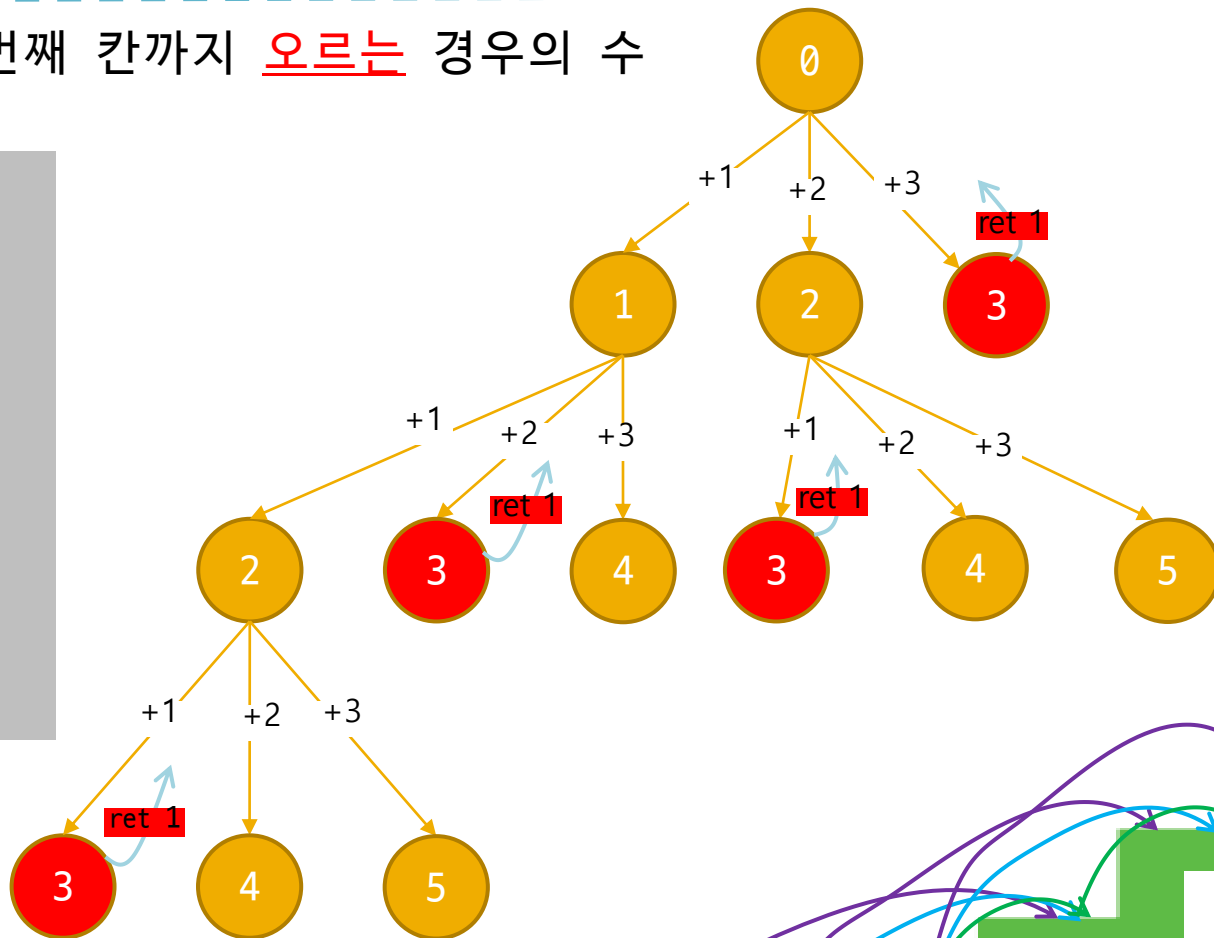
# 계단오르기 - 경우의 수 구하기 - 전역변수 사용

$f(0, 3)$  : 0번에서 3번째 칸까지 오르는 경우의 수

```
def f(s) :  
    if s>e : return 0  
    if s==e : return 1
```

```
    a = f(s+1)  
    b = f(s+2)  
    c = f(s+3)  
    return a + b + c
```

```
e = 3  
print(f(0))
```



```
def f(s) :  
    if s>e : return 0  
    if s==e : return 1  
  
    return f(s+1, e) + f(s+2, e) + f(s+3, e)
```

```
e = 3  
print(f(0))
```

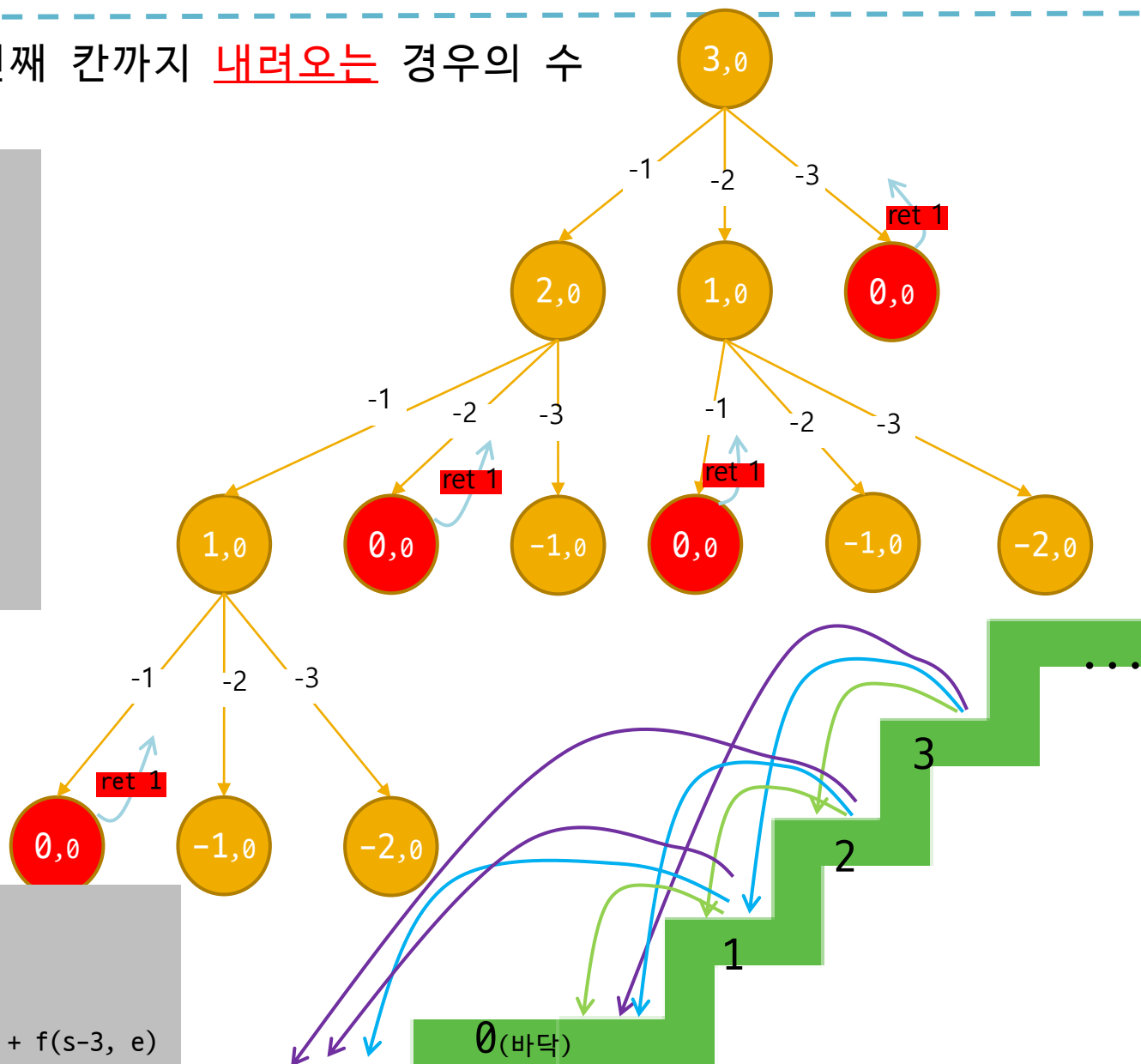
# 계단오르기 - 경우의 수 구하기

$f(3, 1)$  : 0번에서 3번째 칸까지 내려오는 경우의 수

```
def f(s, e) :  
    if s < e : return 0  
    if s == e : return 1
```

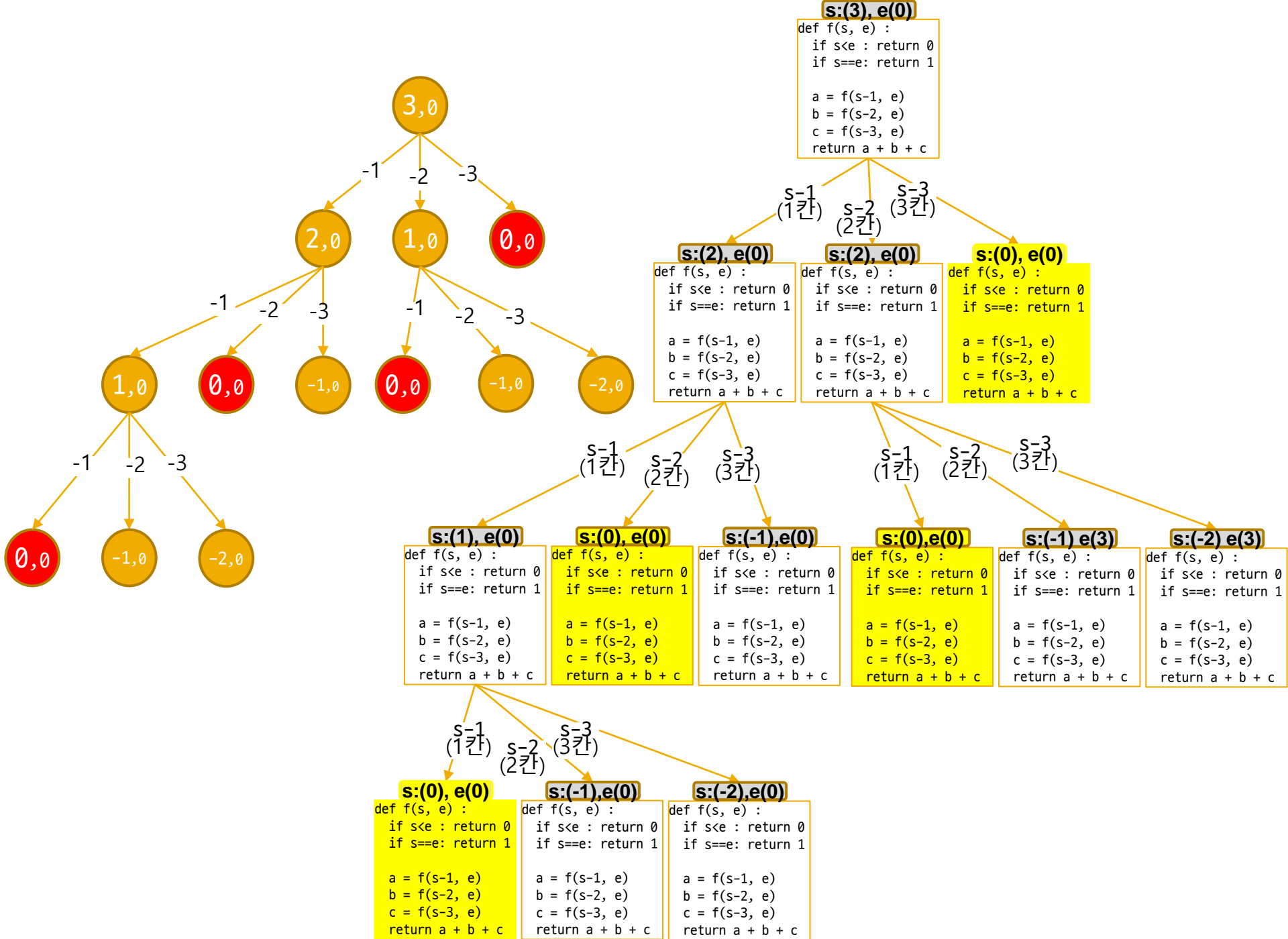
```
    a = f(s-1, e)  
    b = f(s-2, e)  
    c = f(s-3, e)  
    return a + b + c
```

```
print(f(3, 0))
```

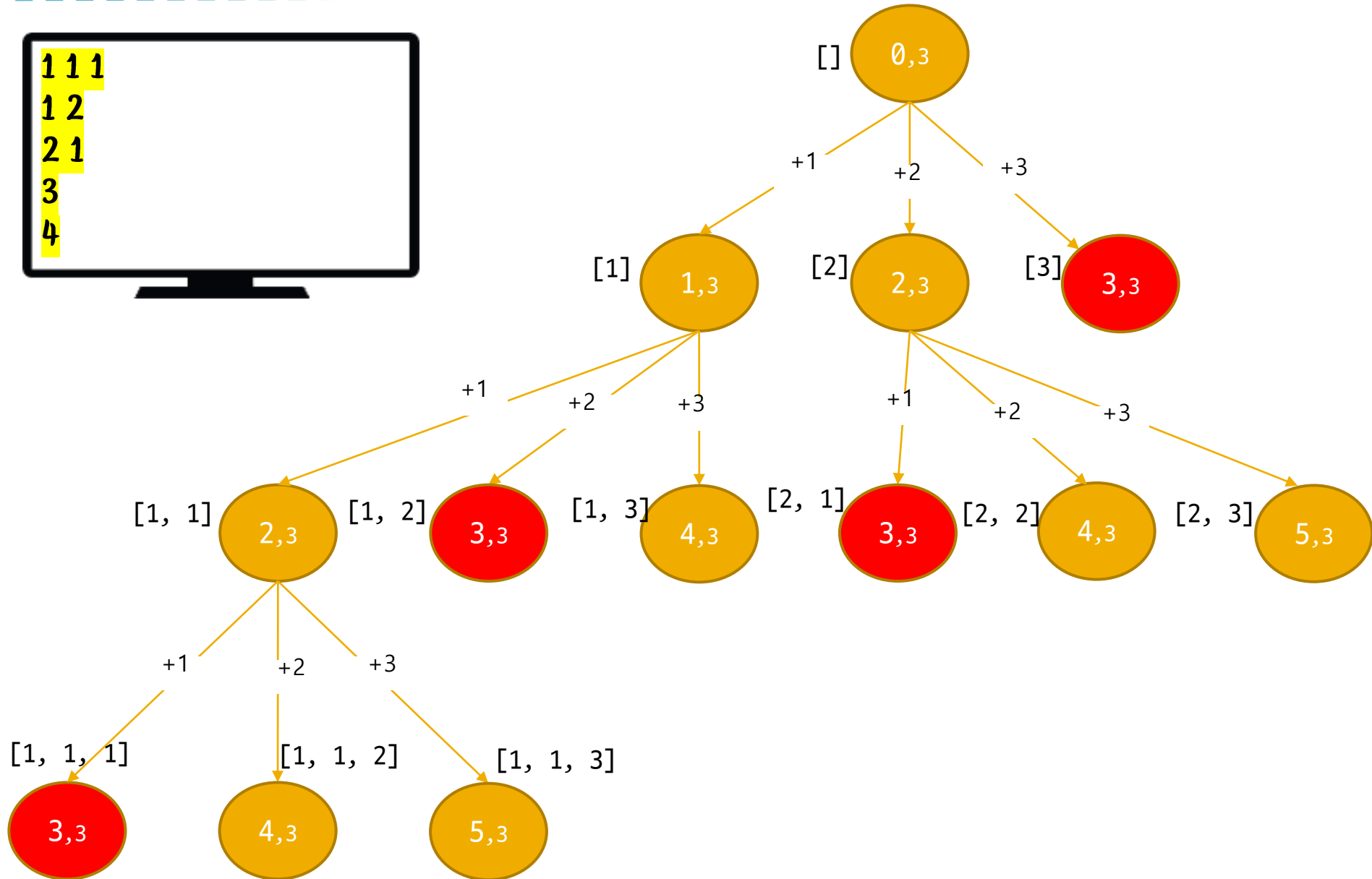
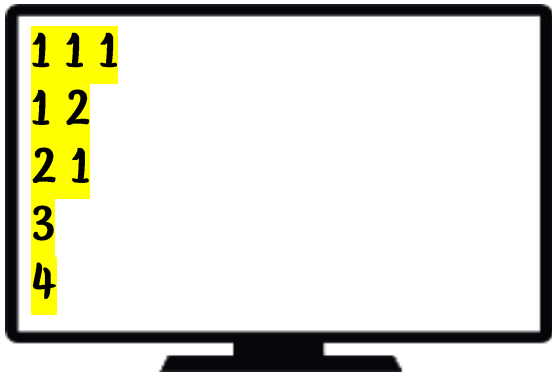


```
def f(s, e) :  
    if s < 0 : return 0  
    if s == 0 : return 1  
  
    return f(s-1, e) + f(s-2, e) + f(s-3, e)
```

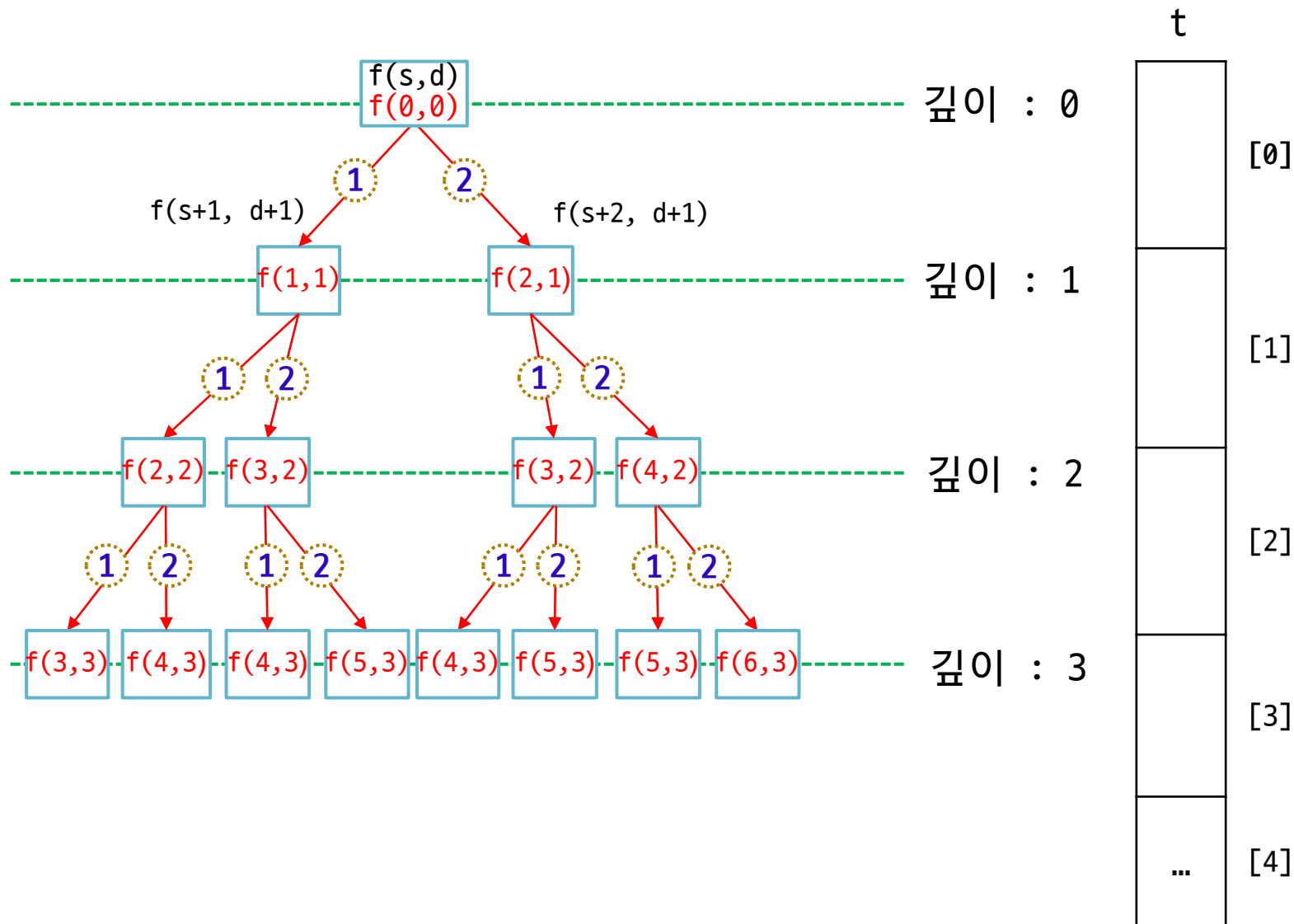
```
print(f(3, 0))
```



# 계단오르기 - 모든 방법을 출력



# 0번(바닥) -> n번 계단까지 오르는 방법(수)



# 문제 AE: 배낭(knapsack)0 - (python)

시간제한: 1.0sec

메모리제한: 128MiB

통과: 19

제출: 406

제출

제출화면

문제리스트

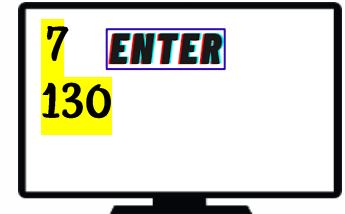
문제수정

TESTDATA

## 문제 설명

지원은 보석들로 가득한 보물섬을 발견했다.  
보석은 3종류인데 각각의 무게와 가치는 아래와 같다.

보물종류	무게	가치
1	2	40
2	3	30
3	5	90



지원의 배낭이 견딜 수 있는 무게가 입력되면, 배낭에 넣을 수 있는 최대 가치를 출력하는 프로그램을 완성하시오.

## 입력/출력 데이터 형식

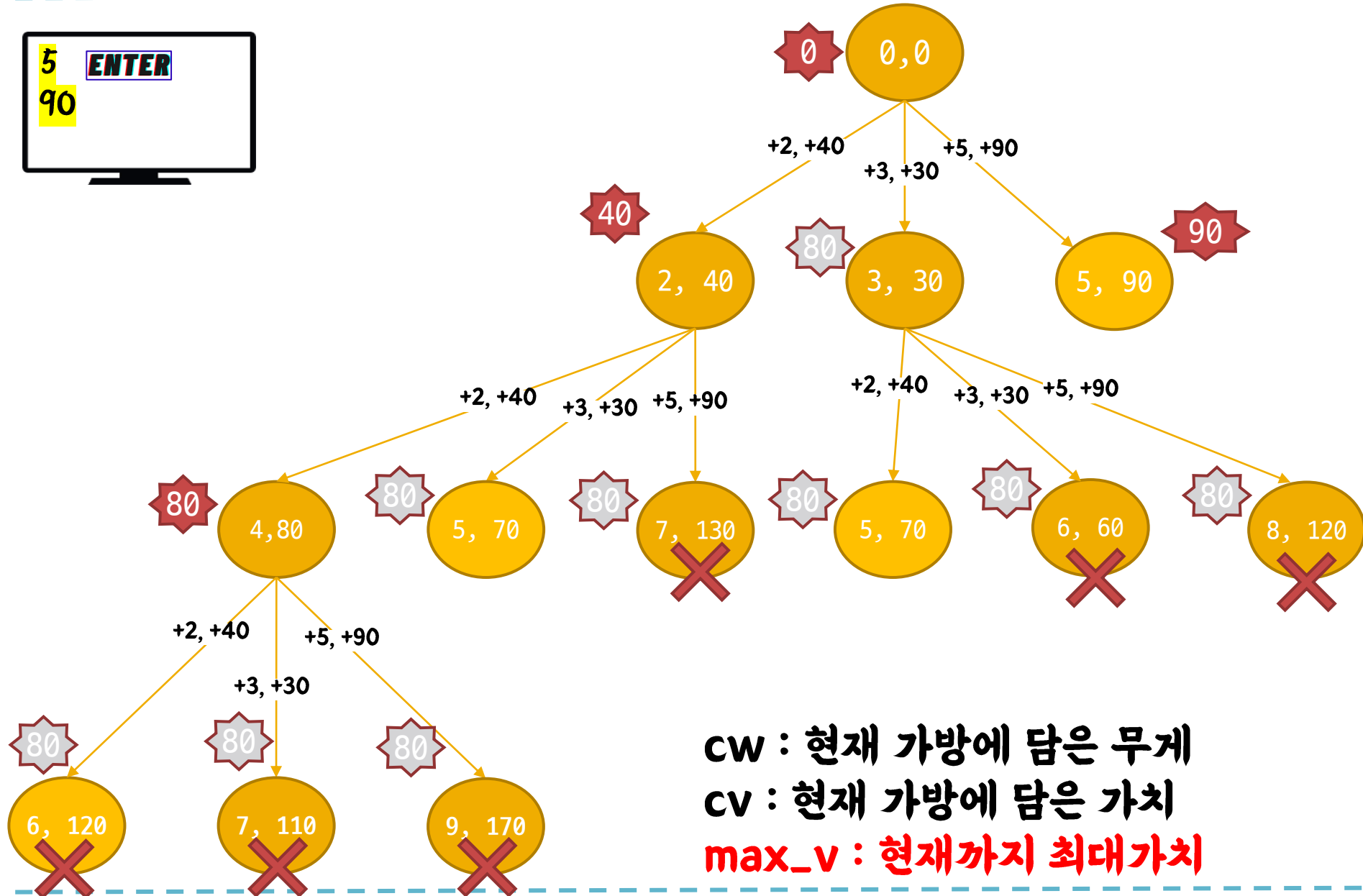
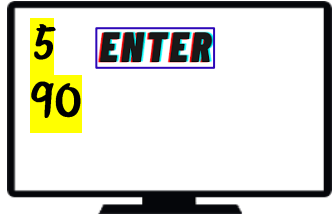
### 입력 데이터 형식

배낭에 담을 수 있는 무게( $w$ )가 입력된다. ( $2 \leq w \leq 30$ )

### 출력 데이터 형식

배낭에 담을 수 있는 최대가치가 출력된다.

# 배낭 - 배낭에 담을 수 있는 최대가치 구하기





# 배낭 - 배낭에 담을 수 있는 최대가치 구하기

