Chapter. 04

재귀 호출

핵심 유형 문제풀이

FAST CAMPUS
ONLINE
유형별 문제풀이

강사. 나동빈

Chapter. 04

재귀 호출(핵심 유형 문제풀이)

I 혼자 힘으로 풀어 보기

피보나치 수: https://www.acmicpc.net/problem/2747

문제 제목: 피보나치 수

문제 난이도: 하(Easy)

문제 유형: 재귀 함수

추천 풀이 시간: 15분

피보나치 수: https://www.acmicpc.net/problem/2747

I 문제 풀이 핵심 아이디어

1. 피보나치 수열의 점화식을 세웁니다.

$$F_0 = 0, F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \ (n \geq 2)$$

2. 재귀 함수를 이용해 문제를 풀 수 있는지 검토해야 합니다.
3. 문제에서 N은 최대 45입니다.

I 문제 풀이 핵심 아이디어

피보나치 수: https://www.acmicpc.net/problem/2747

실패 소스코드

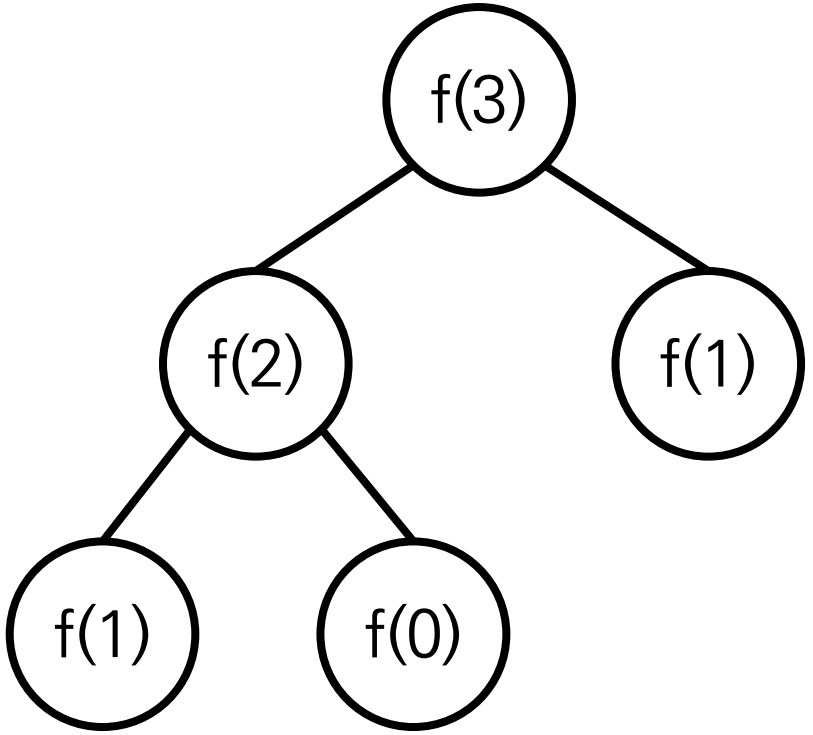
```python
def fibonacci(n):
    if n == 0:
        return 0
    if n == 1:
        return 1
    return fibonacci(n - 1) + fibonacci(n - 2)

print(fibonacci(int(input())))
```

피보나치 수: https://www.acmicpc.net/problem/2747

I 문제 풀이 핵심 아이디어

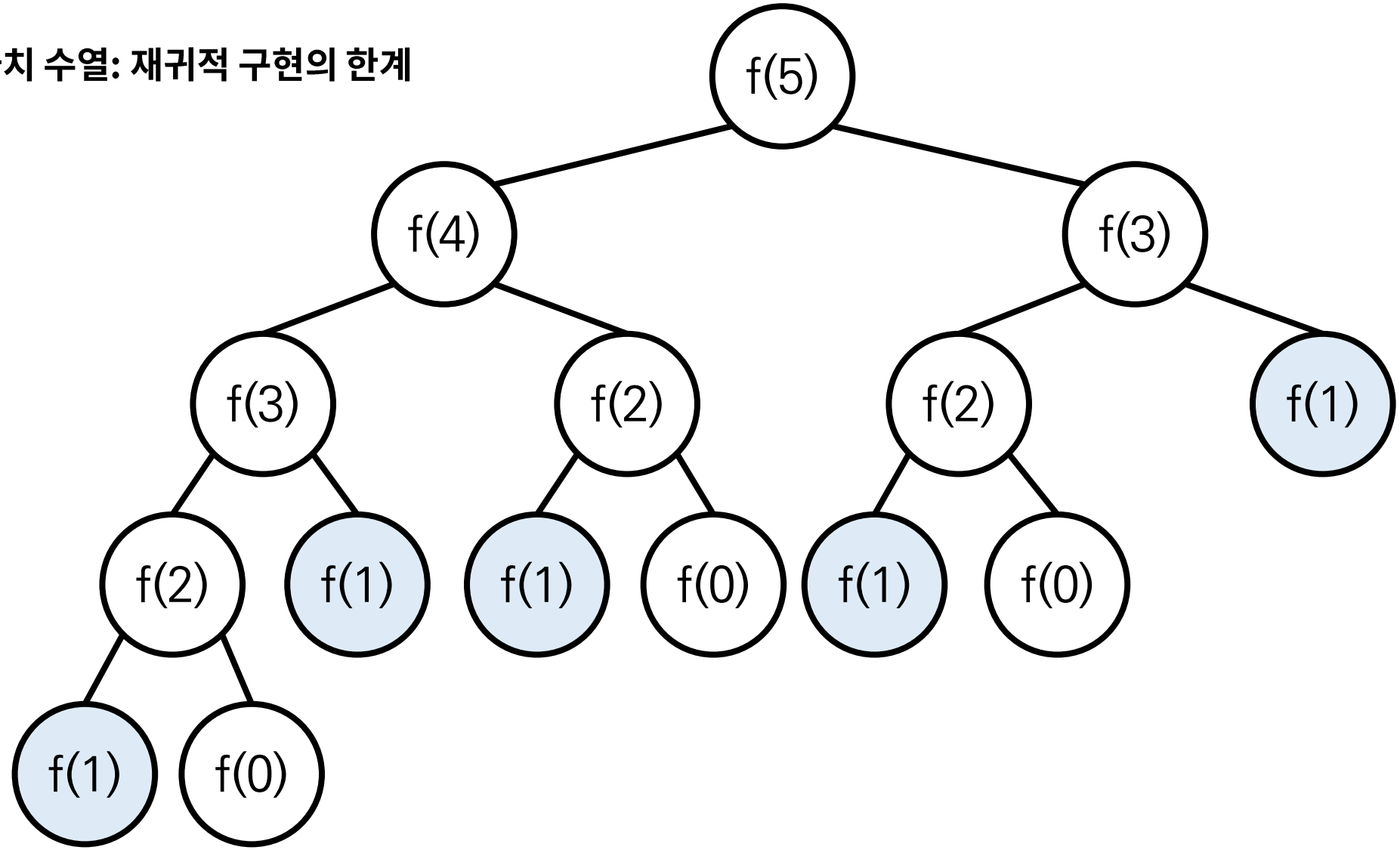
피보나치 수열: 재귀적 구현의 한계



피보나치 수: https://www.acmicpc.net/problem/2747

I 문제 풀이 핵심 아이디어

피보나치 수열: 재귀적 구현의 한계



피보나치 수: https://www.acmicpc.net/problem/2747

| 소스코드

```python
n = int(input())

a, b = 0, 1

while n > 0:
  a, b = b, a + b
  n -= 1

print(a)
```

Z: https://www.acmicpc.net/problem/1074

I 혼자 힘으로 풀어 보기

문제 제목: Z

문제 난이도: 중(Medium)

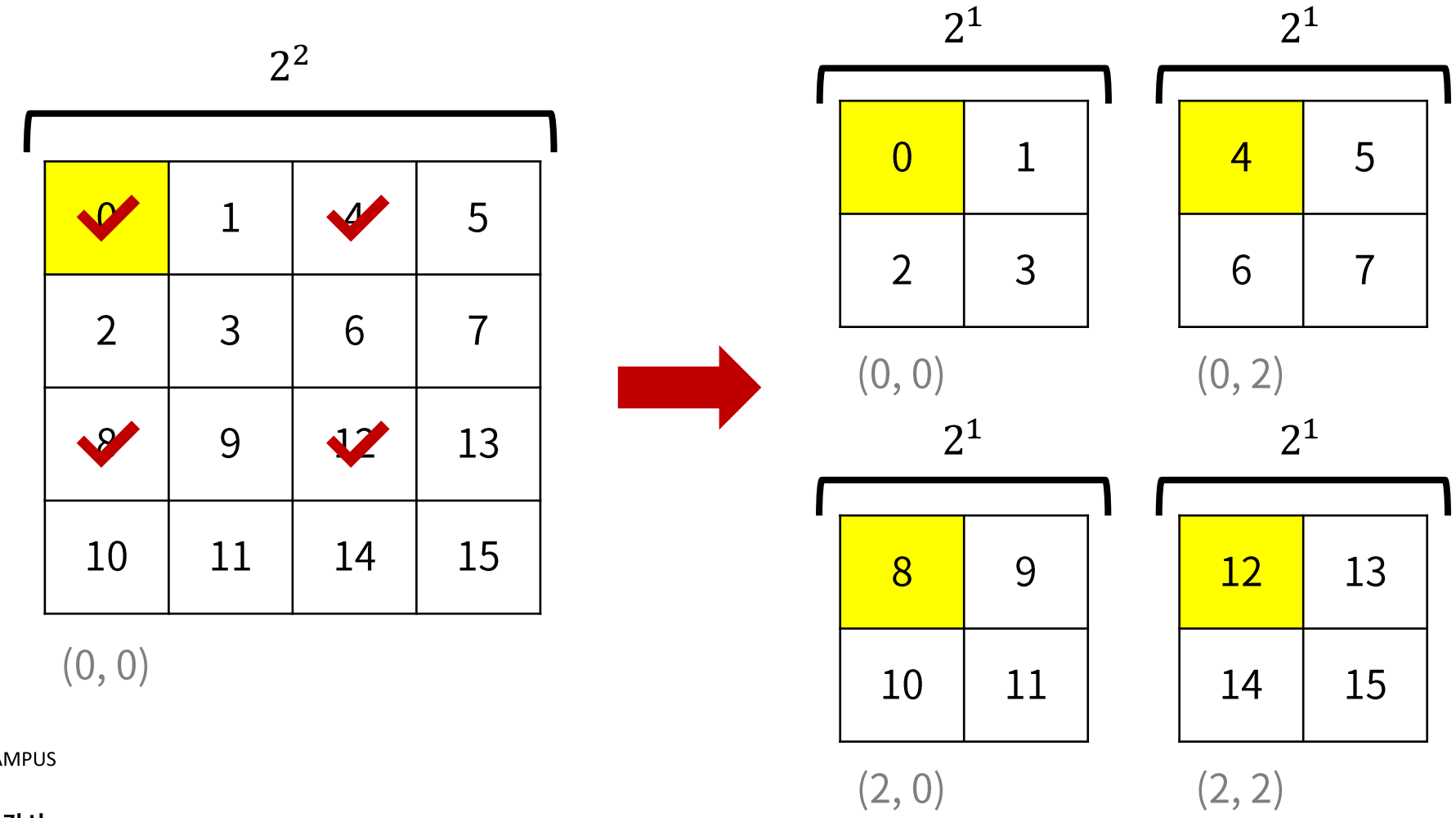
문제 유형: 재귀 함수

추천 풀이 시간: 40분

Z: https://www.acmicpc.net/problem/1074

I 문제 풀이 핵심 아이디어

1. Z 모양을 구성하는 4가지 방향에 대하여 차례대로 재귀적으로 호출합니다.



Z: https://www.acmicpc.net/problem/1074

| 소스코드

```
def solve(n, x, y):
    global result
    if n == 2:
        if x == X and y == Y:
            print(result)
            return
        result += 1
        if x == X and y + 1 == Y:
            print(result)
            return
        result += 1
        if x + 1 == X and y == Y:
            print(result)
            return
        result += 1
        if x + 1 == X and y + 1 == Y:
            print(result)
            return
        result += 1
        return
    solve(n / 2, x, y)
    solve(n / 2, x, y + n / 2)
    solve(n / 2, x + n / 2, y)
    solve(n / 2, x + n / 2, y + n / 2)
```

```
result = 0
N, X, Y = map(int, input().split(' '))
solve(2 ** N, 0, 0)
```

I 혼자 힘으로 풀어 보기

0 만들기: https://www.acmicpc.net/problem/7490

문제 제목: 0 만들기

문제 난이도: 중(Medium)

문제 유형: 재귀 함수

추천 풀이 시간: 40분

0 만들기: https://www.acmicpc.net/problem/7490

I 문제 풀이 핵심 아이디어

1. 자연수 N의 범위(3 <= N <= 9)가 매우 한정적이므로 완전 탐색으로 문제를 해결할 수 있습니다.
2. 수의 리스트와 연산자 리스트를 분리하여 모든 경우의 수를 계산합니다.

0 만들기: https://www.acmicpc.net/problem/7490

I 문제 풀이 핵심 아이디어

수 리스트

1	2	3

연산자 리스트

‘ ’	‘ ’	‘ + ’	‘ ’	‘ – ’	‘ ’
‘ ’	‘ + ’	‘ + ’	‘ + ’	‘ – ’	‘ + ’
‘ ’	‘ – ’	‘ + ’	‘ – ’	‘ – ’	‘ – ’

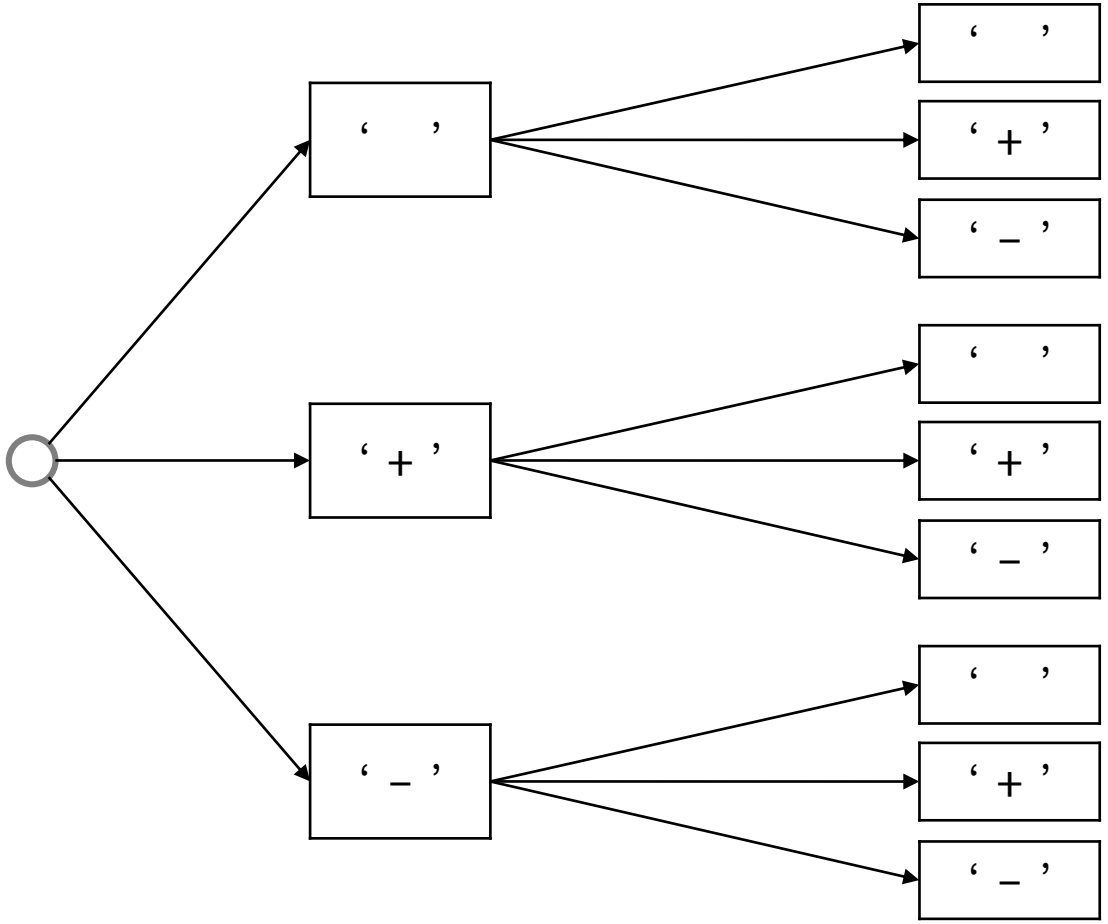
[전체 경우의 수]

1 2 3	1 + 2 3	1 - 2 3
1 2 + 3	1 + 2 + 3	1 - 2 + 3
1 2 - 3	1 + 2 - 3	1 - 2 - 3

0 만들기: https://www.acmicpc.net/problem/7490

I 문제 풀이 핵심 아이디어

1. 가능한 모든 경우를 고려하여 연산자 리스트를 만드는 것이 관건입니다. (재귀 함수 이용)
2. 파이썬의 *eval()* 함수를 이용하여 문자열 형태의 표현식을 계산할 수 있습니다.



0 만들기: https://www.acmicpc.net/problem/7490

| 소스코드

```python
import copy

def recursive(array, n):
    if len(array) == n:
        operators_list.append(copy.deepcopy(array))
        return
    array.append(' ')
    recursive(array, n)
    array.pop()

    array.append('+')
    recursive(array, n)
    array.pop()

    array.append('-')
    recursive(array, n)
    array.pop()
```

```python
test_case = int(input())

for _ in range(test_case):
    operators_list = []
    n = int(input())
    recursive([], n - 1)

    integers = [i for i in range(1, n + 1)]

    for operators in operators_list:
        string = ""
        for i in range(n - 1):
            string += str(integers[i]) + operators[i]
        string += str(integers[-1])
        if eval(string.replace(" ", "")) == 0:
            print(string)
    print()
```