

ICPC 6주차 문제풀이

뮤탈리스크

소프트웨어학과 201921017 이지우

Ideas

- ① 체력이 많이 남은 순서대로 공격
- ② 최대한 빨리 SCV의 수를 줄이기
== 체력이 적은 SCV를 먼저 공격



①, ② 모두 반례가 명확하고
체력이 가장 많고 적은 SCV를 찾는 것이 어렵고 비효율적

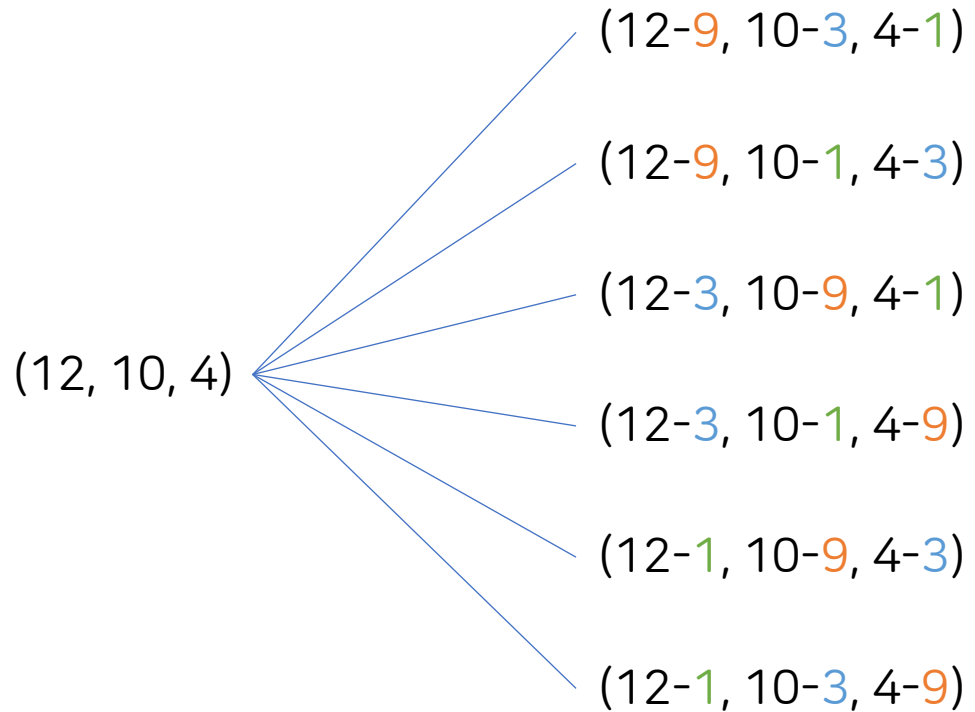
- ③ 각 SCV의 현재 남은 체력에 가장 fit하게 맞는 damage를 입히기



SCV의 체력이 클 때는 어떤 것이 가장 fit한지 구하기 어렵고
정확한 결과도 나오지 않음

(12, 10, 4)의 체력을 가진 SCV를 모두 파괴하는 최소 공격 횟수

④ 가능한 공격 횟수 중 최소 공격 횟수 구하기 : Top-down 방식의 DP



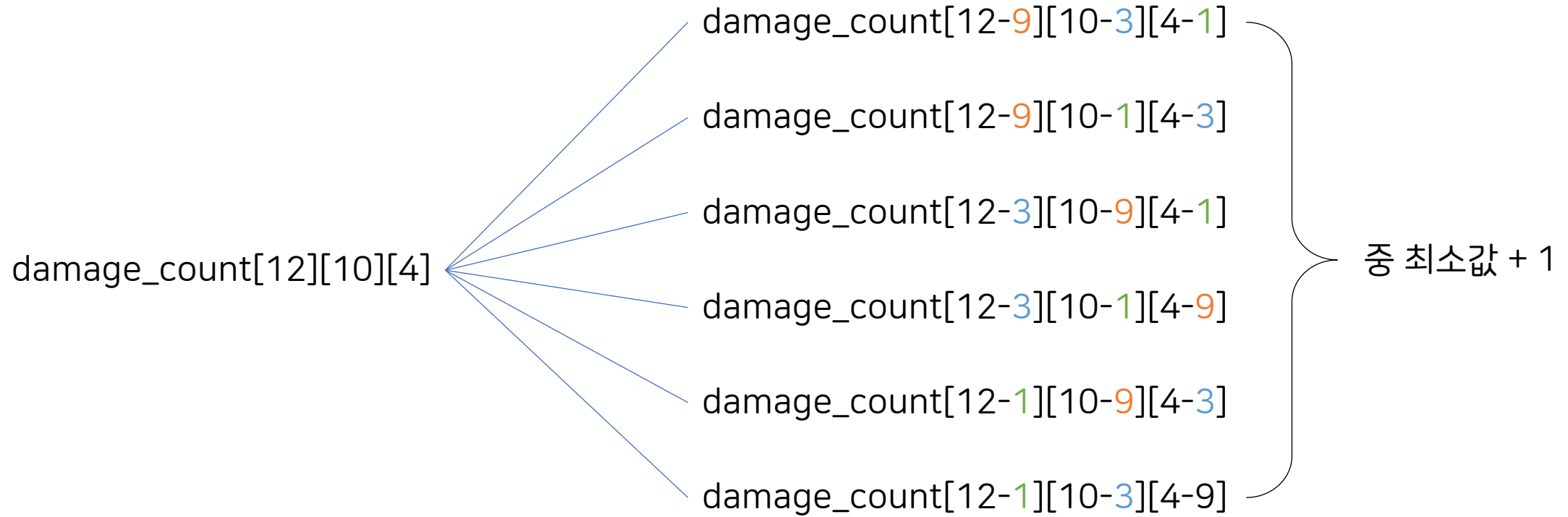
damage_count[X][Y][Z] :

첫번째 SCV의 체력이 X

두번째 SCV의 체력이 Y

세번째 SCV의 체력이 Z일 때의 최소 공격 횟수

④ 가능한 공격 횟수 중 최소 공격 횟수 구하기 : Top-down 방식의 DP



Implementation - 문제풀이

```
if p1 <= 0 and p2 <= 0 and p3 <= 0:
    return 0

if p1 < 0: p1 = 0
if p2 < 0: p2 = 0
if p3 < 0: p3 = 0

if damage_cnt[p1][p2][p3]: # 이미 값이 채워진 경우, 재귀하지 않음
    return damage_cnt[p1][p2][p3]
else:
    damage_cnt[p1][p2][p3] = min(
        attackSCV(p1-9, p2-3, p3-1),
        attackSCV(p1-9, p2-1, p3-3),
        attackSCV(p1-3, p2-9, p3-1),
        attackSCV(p1-3, p2-1, p3-9),
        attackSCV(p1-1, p2-9, p3-3),
        attackSCV(p1-1, p2-3, p3-9)) + 1
    return damage_cnt[p1][p2][p3]
```

```
scv_powers = [0, 0, 0]

n = int(input())


if n == 1: scv_powers[0] = int(input())
elif n == 2: scv_powers[0], scv_powers[1] = map(int, input().split())
else: scv_powers = [*map(int, input().split())]

damage_cnt = [[[0 for _ in range(61)] for _ in range(61)] for _ in range(61)]

print(attackSCV(scv_powers[0], scv_powers[1], scv_powers[2]))
```

SCV의 초기 체력을 담는 scv_powers는 $N(1 \leq N \leq 3)$ 의 값과 상관없이 항상 크기가 3
각 SCV의 체력은 $[1, 60]$ 이므로 체력이 모두 0이 될 때를 고려하여 damage_cnt(61, 61, 61)

Result

28163273	january	 12869	맞았습니다!!	31076 KB	96 ms	Python 3 / 수정	1049 B
----------	---------	--	---------	-------------	-------	------------------	--------

Thank you
