

SIT/2021

뮤탈리스크

- DP -

안태진(taejin7824@gmail.com)

GitHub(github.com/taejin1221)

상명대학교 소프트웨어학과

201821002

Contents

- Problem
- Approach
- Code

Contents

- Problem
- Approach
- Code

Problem

- Summary

- 뮤탈리스크가 3 마리의 SCV를 공격할 때 몇번 공격해야 다 잡을 수 있을까?
- 뮤탈리스크의 공격처럼 첫번째는 9, 두번째는 3 세번째는 1
- 20마리 잡는걸로 다5 문제

뮤탈리스크 2

Diamond V

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람
2 초	512 MB	55	16	10

문제

수빈이는 강호와 함께 스타크래프트 게임을 하고 있다. 수빈이는 뮤탈리스크 1개가 남아있고, 강호는 SCV N개가 남아있다. 각각의 SCV는 남아있는 체력이 주어졌으며, 뮤탈리스크를 공격할 수는 없다. 즉, 이 게임은 수빈이가 이겼다는 것이다. 뮤탈리스크가 공격을 할 때, 한 번에 세 개의 SCV를 공격할 수 있다.

1. 첫 번째로 공격받는 SCV는 체력 9를 잃는다.
2. 두 번째로 공격받는 SCV는 체력 3을 잃는다.
3. 세 번째로 공격받는 SCV는 체력 1을 잃는다.

SCV의 체력이 0 또는 그 이하가 되어버리면, SCV는 그 즉시 파괴된다. 한 번의 공격에서 같은 SCV를 여러 번 공격할 수는 없다. 남아있는 SCV의 체력이 주어졌을 때, 모든 SCV를 파괴하기 위해 공격해야 하는 횟수의 최솟값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 SCV의 수 N ($1 \leq N \leq 20$)이 주어진다. 둘째 줄에는 SCV N개의 체력이 주어진다. 체력은 60보다 적거나 같은 자연수이다.



Contents

- Problem
- Approach
- Code

Approach

- Approach 1
 - SCV 3마리의 체력을 a, b, c 라 할 때 한 번 공격할 수 있는 경우의 수
 1. $(a - 9), (b - 3), (c - 1)$
 2. $(a - 9), (b - 1), (c - 3)$
 3. $(a - 3), (b - 9), (c - 1)$
 4. $(a - 3), (b - 1), (c - 9)$
 5. $(a - 1), (b - 9), (c - 3)$
 6. $(a - 1), (b - 3), (c - 9)$

Approach

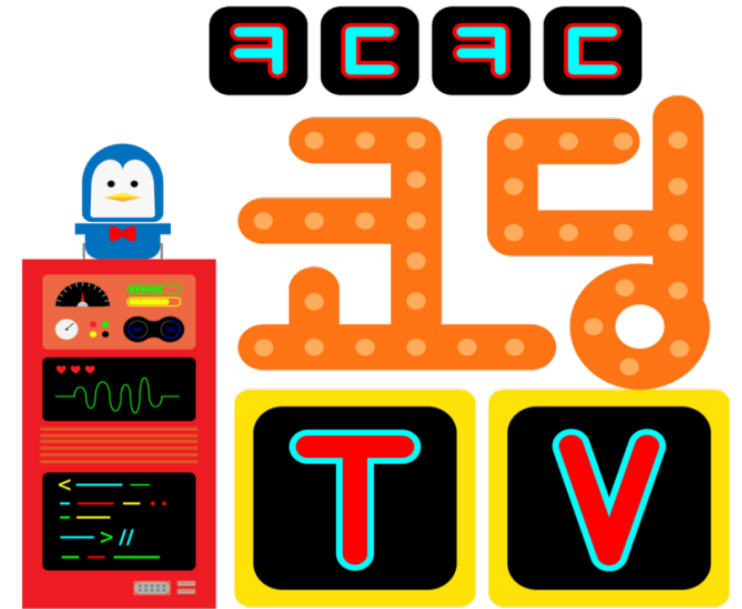
- Approach 1
 - 일단 Bruteforcing 해보자!
 - 최소 공격 횟수를 찾는 것이기 때문에 BFS로 상태 트리를 탐색하자
 - 하지만 1->2나 2->1은 똑같은 결과
 - 따라서 BFS처럼 Bruteforcing을 하면 중복 발생
 - 중복엔?
 - DP!

Approach

- Approach 2
 - DP로 푸려면 상태를 저장해야 하는데 어떤 것을 상태로?
 - SCV의 체력!
 - $dp[a][b][c]$: SCV의 체력이 각각 a, b, c 일 때의 최소 공격 횟수
 - $\therefore dp[a][b][c] = \min(\text{all cases}) + 1$
 - 하지만 Bottom-up일 때의 어려운 점!
 - 공격력이 9, 3, 1 이런 식이기 때문에 예를 들어 $dp[a][b][c] \sim dp[a + 9][b][c]$ 은 전부 $dp[a][b][c] + 1$ 로 채워야 함
 - 처리해야할 반복이 많아짐

Approach

- Approach 2
 - 그래서 Top-down으로 함 ㅋㅋㅋㅋ
 - 생각하기 귀차너... 어려워...



Contents

- Problem
- Approach
- Code

Code

- n 입력 및 초기화
 - min으로 연산할 것이기 때문에 적당히 큰 100으로 초기화

```
int main(void) {  
    int n;  
    cin >> n;  
  
    for ( int i = 0; i < 61; i++ )  
        for ( int j = 0; j < 61; j++ )  
            for ( int k = 0; k < 61; k++ )  
                dp[i][j][k] = 100;
```

Code

- 체력 입력 및 함수 호출
 - 스코프

```
int scv[3] = { 0, };  
for ( int i = 0; i < n; i++ )  
    cin >> scv[i];  
  
dp[0][0][0] = 0;  
cout << solve( scv[0], scv[1], scv[2] ) << '\n';  
  
return 0;  
}
```

Code

- 전역변수 선언
 - 재귀니까!

```
int dp[61][61][61];
```

- 체력이 0보다 작으면 죽은 것 따라서 0으로 초기화

```
int solve( int a, int b, int c ) {  
    if ( a < 0 )  
        a = 0;  
  
    if ( b < 0 )  
        b = 0;  
  
    if ( c < 0 )  
        c = 0;  
}
```

Code

- $dp[a][b][c] = \min(all\ cases) + 1$ 구현
 - 노가다!
 - 머리가 나쁘면 몸이 고생한다!
 - 머리 쓰기 귀찮으니 몸을 귀찮게 하자!
- 만약 dp가 비어있으면 실행

```
if ( dp[a][b][c] == 100 ) {  
    int curr = dp[a][b][c];  
    curr = min( curr, solve( a - 9, b - 3, c - 1 ) + 1 );  
    curr = min( curr, solve( a - 9, b - 1, c - 3 ) + 1 );  
    curr = min( curr, solve( a - 3, b - 9, c - 1 ) + 1 );  
    curr = min( curr, solve( a - 3, b - 1, c - 9 ) + 1 );  
    curr = min( curr, solve( a - 1, b - 9, c - 3 ) + 1 );  
    curr = min( curr, solve( a - 1, b - 3, c - 9 ) + 1 );  
  
    dp[a][b][c] = curr;  
}
```

Code

- 제일 중요한 부분!
 - 이 부분에서 제일 많이 틀림
 - 이 부분에서 실수 하는 학생들 많이 봄
 - 재귀 함수에서 가장 중요한 부분!
 - 많은 사람들이 깜빡하는 부분!
- return!

```
return dp[a][b][c];
```
- ㄷ 人...

감사합니다!
