

DP, 완탐

🕒 Created	@January 10, 2022 10:13 PM
🏷 Tags	
📖 난이도	

우수마을 1949

<https://www.acmicpc.net/problem/1949>

문제

N개의 마을로 이루어진 나라가 있다. 편의상 마을에는 1부터 N까지 번호가 붙어 있다고 하자. 이 나라는 트리(Tree) 구조로 이루어져 있다. 즉 마을과 마을 사이를 직접 잇는 N-1개의 길이 있으며, 각 길은 방향성이 없어서 A번 마을에서 B번 마을로 갈 수 있다면 B번 마을에서 A번 마을로 갈 수 있다. 또, 모든 마을은 연결되어 있다. 두 마을 사이에 직접 잇는 길이 있을 때, 두 마을이 인접해 있다고 한다.

이 나라의 주민들에게 성취감을 높여 주기 위해, 다음 세 가지 조건을 만족하면서 N개의 마을 중 몇 개의 마을을 '우수 마을'로 선정하려고 한다.

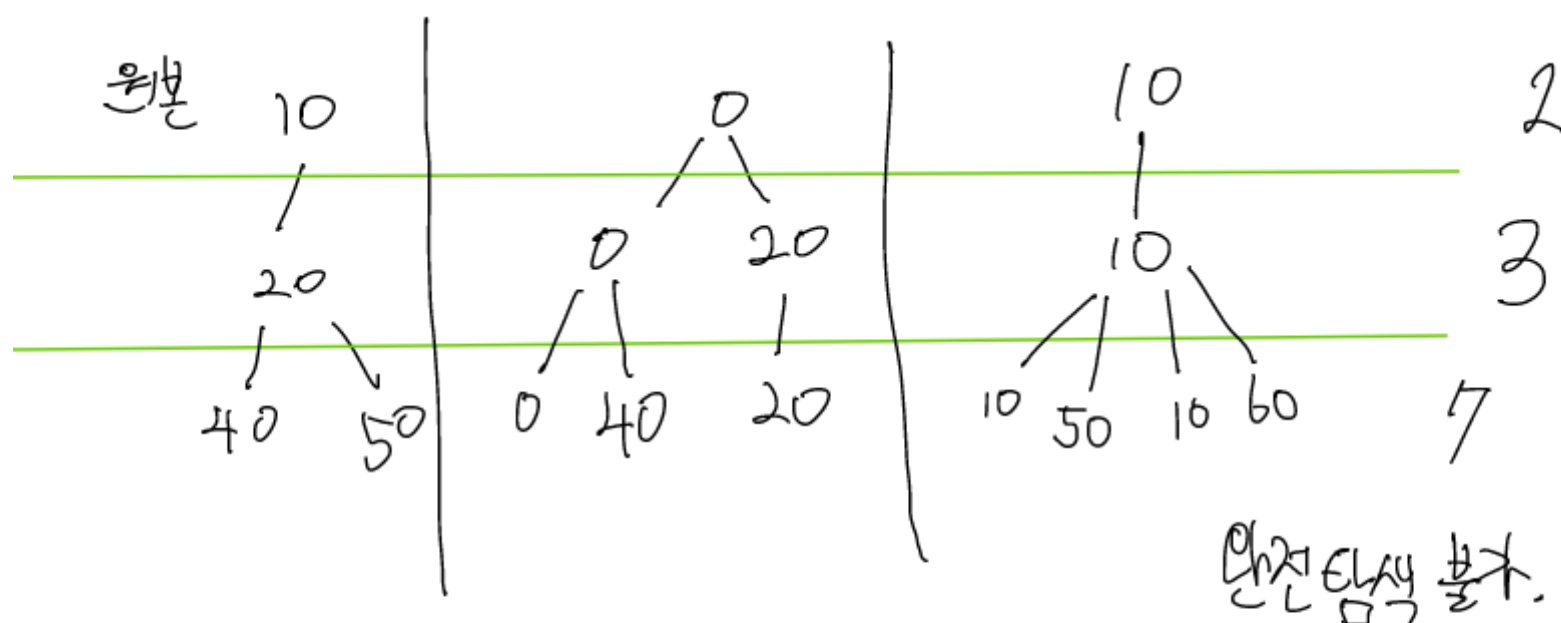
1. '우수 마을'로 선정된 마을 주민 수의 총 합을 최대한 해야 한다.
2. 마을 사이의 충돌을 방지하기 위해서, 만일 두 마을이 인접해 있으면 두 마을을 모두 '우수 마을'로 선정할 수는 없다. 즉 '우수 마을'끼리는 서로 인접해 있을 수 없다.
3. 선정되지 못한 마을에 경각심을 불러일으키기 위해서, '우수 마을'로 선정되지 못한 마을은 적어도 하나의 '우수 마을'과는 인접해 있어야 한다.

각 마을 주민 수와 마을 사이의 길에 대한 정보가 주어졌을 때, 주어진 조건을 만족하도록 '우수 마을'을 선정하는 프로그램을 작성하시오.

완전 탐색

BFS를 활용.

우수 마을 선정 O, 우수 마을 선정 X를 큐에 넣는다.



우수 마을 = 1
우수 마을 X = 2

시간 초과.

N은 10_000 으로 시간초과 예상.

DP

2차원 배열을 사용해서 노드만 거쳐가자.

```
private static void solution() {
    dfs(0, 1);
    System.out.println(Math.max(dp[1][0], dp[1][1]));
}

private static void dfs(int prev, int cur) {
    // 현재 노드를 우수 마을로 선정[1]하거나, 선정하지 않거나[0]
    // dp[cur][0] = 0;
    dp[cur][1] = population[cur];

    for (int next : graph[cur]) {
        if (prev == next) continue;
        // 리프 노드까지 내려간다.
        dfs(cur, next);

        // 점화식
        // 현재 노드를 우수 마을로 선정하지 않는다면, 인접 노드는 우수마을이거나 아닌 경우 중 큰 값이어야 한다.
        // 현재 노드를 우수 마을로 선정한다면, 인접 노드는 반드시 우수마을이 아니어야 한다.
        dp[cur][0] += Math.max(dp[next][0], dp[next][1]);
        dp[cur][1] += dp[next][0];
    }
}
```

시간 복잡도 $O(V + E)$ 로 수행할 수 있다.

기타리스트 1495

기타리스트 성공



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	14629	5179	3999	34.501%

문제

Day Of Mourning의 기타리스트 강토는 다가오는 공연에서 연주할 N개의 곡을 연주하고 있다. 지금까지 공연과는 다른 공연을 보여주기 위해서 이번 공연에서는 매번 곡이 시작하기 전에 볼륨을 바꾸고 연주하려고 한다.

먼저, 공연이 시작하기 전에 각각의 곡이 시작하기 전에 바꿀 수 있는 볼륨의 리스트를 만들었다. 이 리스트를 V라고 했을 때, $V[i]$ 는 i번째 곡을 연주하기 전에 바꿀 수 있는 볼륨을 의미한다. 항상 리스트에 적힌 차이로만 볼륨을 바꿀 수 있다. 즉, 현재 볼륨이 P이고 지금 i번째 곡을 연주하기 전이라면, i번 곡은 $P+V[i]$ 나 $P-V[i]$ 로 연주해야 한다. 하지만, 0보다 작은 값으로 볼륨을 바꾸거나, M보다 큰 값으로 볼륨을 바꿀 수 없다.

곡의 개수 N과 시작 볼륨 S, 그리고 M이 주어졌을 때, 마지막 곡을 연주할 수 있는 볼륨 중 최댓값을 구하는 프로그램을 작성하시오. 모든 곡은 리스트에 적힌 순서대로 연주해야 한다.

입력

첫째 줄에 N, S, M이 주어진다. ($1 \leq N \leq 50$, $1 \leq M \leq 1,000$, $0 \leq S \leq M$) 둘째 줄에는 각 곡이 시작하기 전에 줄 수 있는 볼륨의 차이가 주어진다. 이 값은 1보다 크거나 같고, M보다 작거나 같다.

출력

첫째 줄에 가능한 마지막 곡의 볼륨 중 최댓값을 출력한다. 만약 마지막 곡을 연주할 수 없다면 (중간에 볼륨 조절을 할 수 없다면) -1을 출력한다.

모든 경우의 수를 다 봐야 한다.

입력의 수가 최대 50이니 시간 복잡도는 $O(2^{50} (2^{50} + 1) / 2)$ 단순 완전탐색 불가능



각 트리 레벨에 중복 볼륨을 제거하자 각 레벨의 볼륨을 Set으로 저장해서 해결.

홀수 홀릭 호석 20164

홀수 홀릭 호석

성공 전체 채점



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	512 MB	632	433	341	70.021%

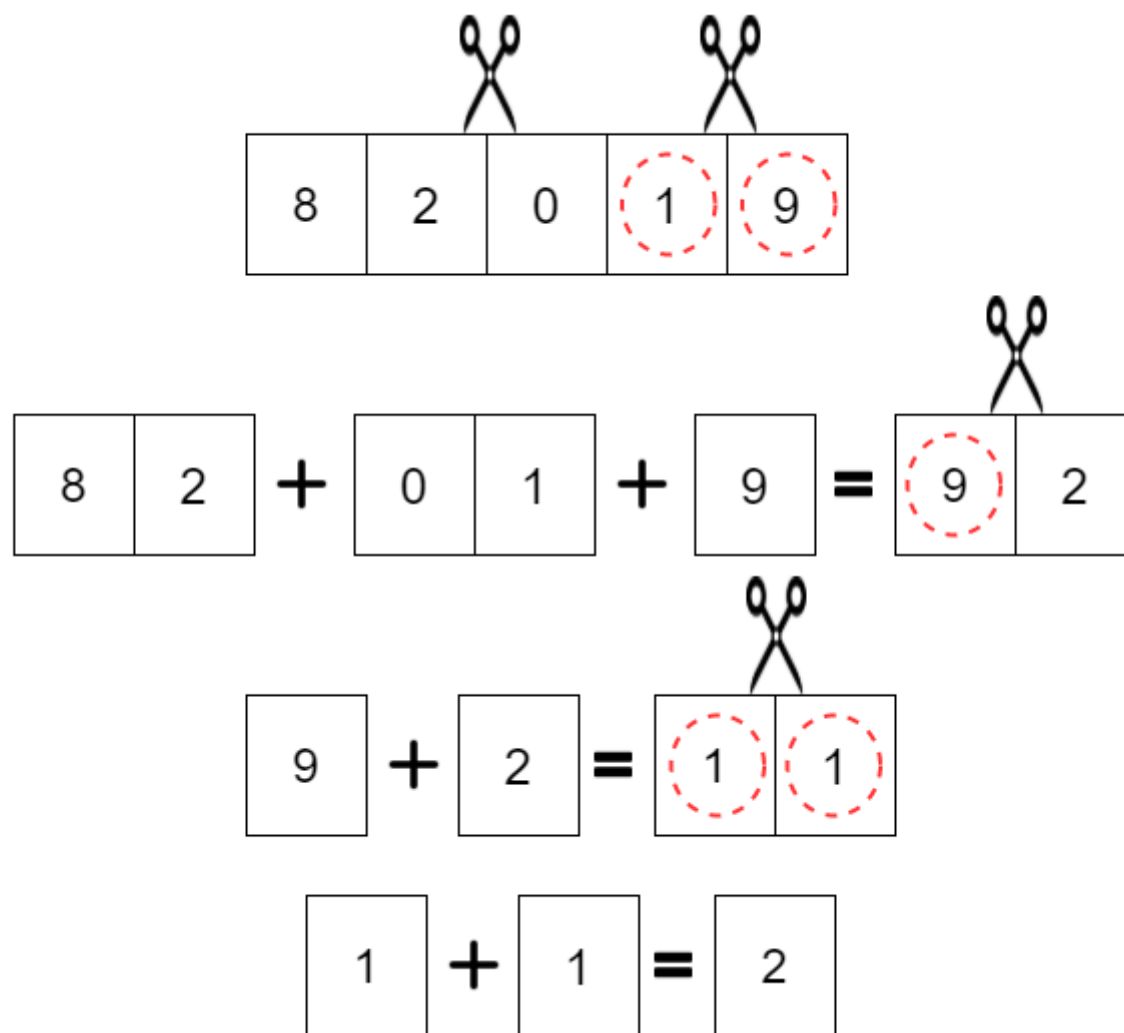
문제

호석이는 짝수랑 홀수 중에서 이니셜이 같은 홀수를 더 좋아한다. 운전을 하던 호석이는 앞차의 번호판이 홀수로 가득할 때 사랑스러움을 느낄 정도이다. 전화번호도 홀수만 있고 싶다. 그렇게 홀수 홀릭에 빠진 호석이는 가지고 있는 수 N 을 일련의 연산을 거치면서, 등장하는 숫자들에서 홀수를 최대한 많이 많이 보고 싶다.

하나의 수가 주어졌을 때 호석이는 한 번의 연산에서 다음과 같은 순서를 거친다.

- 수의 각 자리 숫자 중에서 홀수의 개수를 종이에 적는다.
- 수가 한 자리이면 더 이상 아무것도 하지 못하고 종료한다.
- 수가 두 자리이면 2개로 나눠서 합을 구하여 새로운 수로 생각한다.
- 수가 세 자리 이상이면 임의의 위치에서 끊어서 3개의 수로 분할하고, 3개를 더한 값을 새로운 수로 생각한다.

호석이는 연산이 종료된 순간에 종이에 적힌 수들을 모두 더한다. 그렇게 최종적으로 얻은 수를 최종값이라고 하자. 예를 들어, 시작하는 수가 82019 라고 하자. 그럼 아래와 같이 나누게 되면 5개의 홀수를 볼 수 있기 때문에, 최종값이 5가 된다.



완전탐색

시간 복잡도: O(9개의 숫자 중 3개를 뽑는 경우의 수) $9 * 8 * 7 / 3 * 2 * 1 = 84$

코드 살펴보기