

점프와 순간 이동

이 문제를 선택한 이유는...

Level 3

프로그래밍 언어

- ☐ C
- ☐ C++
- ☐ C#
- ☐ Go
- ☒ Java
- ☐ JavaScript
- ☐ Kotlin
- ☐ Python2
- ☐ Python3
- ☐ Ruby
- ☐ Scala
- ☐ Swift

문제 모음

- ☐ 2020 KAKAO BLIND RECRUITMENT
- ☐ 2019 KAKAO BLIND RECRUITMENT
- ☐ 2018 KAKAO BLIND RECRUITMENT
- ☐ 2017 카카오프렌즈
- ☐ 2017 카카오프렌즈 본선
- ☐ 찾아라 프로그래밍 마에스터
- ☐ 2017 팀스타운
- ☐ 서머코딩/윈터코딩(~2018)
- ☐ 서머코딩/윈터코딩(2019)

코딩테스트 고득점 Kit SQL 고득점 Kit 모든 문제

가장 큰 정사각형 찾기

연습문제 • 1670명 완료



올바른 괄호

연습문제 • 2873명 완료



땅따먹기

연습문제 • 1573명 완료



숫자의 표현

연습문제 • 3020명 완료



최솟값 만들기

연습문제 • 3590명 완료



행렬의 곱셈

연습문제 • 2315명 완료



단체사진 찍기

2017 카카오프렌즈 본선 • 152명 완료



다음 큰 숫자

연습문제 • 3334명 완료



폰켓몬

찾아라 프로그래밍 마에스터 • 1030명 완료



최댓값과 최솟값

연습문제 • 4067명 완료



피보나치 수

연습문제 • 5394명 완료



JadenCase 문자열 만들기

연습문제 • 1927명 완료



lv2 첫방문을 환영해주는 숫자 문제들... 너희들 말고... 어디 재밌는 친구 없니?

문제 모음

- ☐ 2020 KAKAO BLIND RECRUITMENT
- ☐ 2019 KAKAO BLIND RECRUITMENT
- ☐ 2018 KAKAO BLIND RECRUITMENT
- ☐ 2017 카카오코드 예선
- ☐ 2017 카카오코드 본선
- ☐ 찾아라 프로그래밍 마에스터
- ☐ 2017 팀스타운
- ☐ 서머코딩/윈터코딩(~2018)
- ☐ 서머코딩/윈터코딩(2019)

코딩테스트 고득점 Kit SQL 고득점 Kit 모든 문제



최솟값 만들기

연습문제 • 3590명 완료



행렬의 곱셈

연습문제 • 2315명 완료



N개의 최소공배수

연습문제 • 2475명 완료



소수 만들기

서머코딩/윈터코딩(~2018) • 1063명 완료



피보나치 수

연습문제 • 5394명 완료



JadenCase 문자열 만들기

연습문제 • 1927명 완료



짝지어 제거하기

2017 팀스타운 • 744명 완료



점프와 순간 이동

서머코딩/윈터코딩(~2018) • 885명 완료



형..나 찾아..?

이전 1 2 3 다음

우오오오오!!! 너이놈... 구석에 숨어 있었구나... 이리와~ 안아줄게~ //아무무 음성지원



저는 처음 이 문제 제목을 보고 이런
이미지를 연상 했습니다.

그러나...
아니...
역시나...

문제 설명

OO 연구소는 한 번에 K 칸을 앞으로 점프하거나, (현재까지 온 거리) x 2 에 해당하는 위치로 순간이동을 할 수 있는 특수한 기능을 가진 아이언 슈트를 개발하여 판매하고 있습니다. 이 아이언 슈트는 건전지로 작동되는데, 순간이동을 하면 건전지 사용량이 줄지 않지만, 앞으로 K 칸을 점프하면 K 만큼의 건전지 사용량이 듭니다. 그러므로 아이언 슈트를 착용하고 이동할 때는 순간 이동을 하는 것이 더 효율적입니다. 아이언 슈트 구매자는 아이언 슈트를 착용하고 거리가 N 만큼 떨어져 있는 장소로 가려고 합니다. 단, 건전지 사용량을 줄이기 위해 점프로 이동하는 것은 최소로 하려고 합니다. 아이언 슈트 구매자가 이동하려는 거리 N이 주어졌을 때, 사용해야 하는 건전지 사용량의 최소값을 return하는 solution 함수를 만들어 주세요.

예를 들어 거리가 5만큼 떨어져 있는 장소로 가려고 합니다.
아이언 슈트를 입고 거리가 5만큼 떨어져 있는 장소로 갈 수 있는 경우의 수는 여러 가지입니다.

- 처음 위치 0 에서 5 칸을 앞으로 점프하면 바로 도착하지만, 건전지 사용량이 5 만큼 듭니다.
- 처음 위치 0 에서 2 칸을 앞으로 점프한 다음 순간이동 하면 (현재까지 온 거리 : 2) x 2에 해당하는 위치로 이동할 수 있으므로 위치 4로 이동합니다. 이때 1 칸을 앞으로 점프하면 도착하므로 건전지 사용량이 3 만큼 듭니다.
- 처음 위치 0 에서 1 칸을 앞으로 점프한 다음 순간이동 하면 (현재까지 온 거리 : 1) x 2에 해당하는 위치로 이동할 수 있으므로 위치 2로 이동됩니다. 이때 다시 순간이동 하면 (현재까지 온 거리 : 2) x 2 만큼 이동할 수 있으므로 위치 4로 이동합니다. 이때 1 칸을 앞으로 점프하면 도착하므로 건전지 사용량이 2 만큼 듭니다.

위의 3가지 경우 거리가 5만큼 떨어져 있는 장소로 가기 위해서 3번째 경우가 건전지 사용량이 가장 적으므로 답은 2가 됩니다.

제한 사항

- 숫자 N: 1 이상 10억 이하의 자연수
- 숫자 K: 1 이상의 자연수

입출력 예

| N | result |
|---|--------|
| 5 | 2 |

문제 설명을 읽어가며... 서서히...
우뇌가 멈추고 좌뇌가 작동하기 시작 했다.

그럼 이제
시작해볼까요~?

지금 부터 건전지를 최대한 절약하면서
멀~리 이동해야 하는 알고리즘 문제를
풀어보겠습니다.

*알고리즘 설계 과정

*생각나는 대로 적고 공식 찾기

```
1 1 — 2의제곱 2/4/8/16/32/64/128/256/512/1024/2048/4092/8184/16368/
2 1 1 — 힌트 발견 2의제곱이 1점프로 거리 도달 가능하다면 3의 제곱으로는 2번의 점프로 도달가능
3 2 1 — 5의 제곱도 2번
4 1 2 — 반환해야 하는값은 최소의 건전지 소모값
5 2 2 — !! 목표!! = 가장 적은 리소스를 사용하여 최소 점프 사용량으로 거리에 도달하는 로직 설계 및 개발
6 2 2 — 여기서 진행에 변동이 생김 기본 점프1번후 tp1번 거리 2를 만든후 점프1번추가 3을 만든후 tp로 *2를하여 거리 6로 이동
7 2 3 — 여기서는 기본 점프 1번 + tp3번사용하여 2의 3제곱으로 거리 8로 이동 그후 뒤로 점프 한번 거리 7로 이동
8 1 3 — 기본 1점프후 2의 3제곱 거리 8이동
9 2 3 — 기본 1점프후 2의 3제곱 점프1번 거리 9이동
10 2 3 — 기본 1점프후 2의 2제곱 거리 4만든후 점프1번 거리 5 만든후 tp한번 *2하여 거리 10이동
11 3 3 — 기본 1점프후 2의 1제곱 1점프 추가 거리 3만든후 tp2번 사용 거리 12만든후 뒤로 점프 한번 거리 11이동
12 2 3
13 3 3 —첫번째 규칙 발견 tp를 사용하면 짝수가된다. 2의 제곱근에 해당하는 수는 수의 크기에 상관없이 기본 점프 1번으로 tp사용으로 접근 가능하다
14 2 4 —두번째 규칙 14라는 수는 7의 2제곱이기에 7의 tp한번을 추가해준다면 도달가능하다. 건전지 소모 X
15 2 4 —세번째 규칙앞뒤 비교 하여 건전지 소모값 비교 2의 제곱근에 해당하는 수는 단 1번의 점프 소모 16은 2의 제곱근 16에서 백점프 즉 2의점프사용
16 1 5 — 2의제곱 점프1
17 2 5 — 심플하게 생각할것도 없이 2의 4제곱 - 1 이니 16에서 백점프
18 2 4 — 9의 2제곱이기에 9의 jp tp 에 tp 를 +1
19 3 4 — 10의 2제곱에 -1 해주면 되기에 10의 jp tp에
20 2 4 — 10의 제곱
```

n jp tp

*googling 을 통해 다른 분들의 해결방법 찾아보기

2. 해결 방법

1. 최대거리를 계속 2로나눔
2. 2로 안나눠지면 -1하고 1이 될 때 까지 2로 나눔
3. -1할 때 마다 건전지++
4. 마지막 1이 됐을 때 건전지++하고 끝

첫단계 규칙을 찾는 과정 입니다.

신용기님의 알고리즘 발표(124나라의 숫자(2진법 응용 3진법적용))를

보고 문제 풀이 과정에서 많은 도움이 되었습니다.

그.러.나. 입력값(거리)을 1부터 늘려나갈 생각으로만

문제를 해결 하려 했기에 풀리지 않았습니다.

구글링을 하고, 방법을 찾았습니다. 방법을 읽는 순간...



1부터 계산 하던것을 입력값에서 부터
시작 하는 방법으로 문제를 풀어보니
오히려 빠르게 해결할수 있었습니다!

그럼 이제 코드를 보시 겠습니다.

programmers

FEED

BACK

