

1697 숨바꼭질

2021년 4월 9일 금요일 오후 5:53

1 1697번

제출

맞은 사람

스코딩

재채점 결과

채점 현황

내 제출

강의 ▾

질문 검색

숨바꼭질

출처

다국어

분류

☆

한국어 ▾

1 Silver I

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	105304	29472	18333	24.997%

문제

수빈이는 동생과 숨바꼭질을 하고 있다. 수빈이는 현재 점 $N(0 \leq N \leq 100,000)$ 에 있고, 동생은 점 $K(0 \leq K \leq 100,000)$ 에 있다. 수빈이는 걷거나 순간이동을 할 수 있다. 만약, 수빈이의 위치가 X 일 때 걷는다면 1초 후에 $X-1$ 또는 $X+1$ 로 이동하게 된다. 순간이동을 하는 경우에는 1초 후에 $2 \times X$ 의 위치로 이동하게 된다.

수빈이와 동생의 위치가 주어졌을 때, 수빈이가 동생을 찾을 수 있는 가장 빠른 시간이 몇 초 후인지 구하는 프로그램을 작성하시오.

입력

첫 번째 줄에 수빈이가 있는 위치 N 과 동생이 있는 위치 K 가 주어진다. N 과 K 는 정수이다.

출력

수빈이가 동생을 찾는 가장 빠른 시간을 출력한다.

예제 입력 1 복사

5 17

예제 출력 1 복사

4

힌트

수빈이가 5-10-9-18-17 순으로 가면 4초만에 동생을 찾을 수 있다.

```
from collections import deque
```

```
#점 생성
```

```
temp = list(map(int, input().split()))
```

```
# 수빈 위치
```

```
n = temp[0]
```

```
# 동생 위치
```

```
k = temp[1]
```

```
# 방문 위치
```

```
visit = [0] * 100001
```

```
loc = n
```

```
# bfs
```

```
def bfs(visit, start):
```

```
    queue = deque()
```

```
    queue.append(start)
```

```
    # 같은 위치일 경우 예외처리
```

```
    if start == k:
```

```
        return visit[k]
```

```
    while queue:
```

```
        tmp = queue.popleft()
```

```
        # +1 처리 -> tmp+1가 0이거나 tmp+1에 tmp보다 방문할 수가 없을 때
```

```
        tmp+1 <= 100000 and visit[tmp+1] == 0:
```

```
            queue.append(tmp+1)
```

```
            visit[tmp+1] = visit[tmp]+1
```

```
        # -1 처리
```

```
        if tmp-1 >= 0 and visit[tmp-1] == 0:
```

```
            queue.append(tmp-1)
```

```
            visit[tmp-1] = visit[tmp]+1
```

```
        ...
```

같은 위치면 3초301410

Queue에서 dequeue

Queue에 들어간
visit=0
처리를
Queue에 들어간
visit=1
처리

visit을 방문
작을

```

# -1 처리
if tmp-1 >= 0 and visit[tmp-1] == 0:
    queue.append(tmp-1)
    visit[tmp-1] = visit[tmp]+1

# *2 처리
if tmp*2 <= 100000 and visit[tmp*2] == 0:
    queue.append(tmp*2)
    visit[tmp*2] = visit[tmp]+1

# 찾을 경우
if visit[k] != 0:
    return visit[k]
print(bfs(visit, n))

```