

백준 1697 숨바꼭질

202020988 조아영

001 문제 소개

002 아이디어

003 코드 설명

문제 소개

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.

아이디어

현재위치 X에서 수빈이가
이동할 수 있는 위치인
X-1, X, 2*X 를 탐색

- ⇒ BFS 이용
- ⇒ 배열에 각 위치마다 몇번의
이동이 필요한지 저장

단계 마다 visited 배열 일부분 출력 결과

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 2, 1, 2, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 2, 1, 2, 1, 2, 2, 0, 1, 0, 2, 0, 0, 0]
[0, 0, 0, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 0, 0, 0]
[0, 0, 3, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 0, 0, 0]
[0, 0, 3, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 0, 0, 0]
[0, 0, 3, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 0, 3, 0]
[0, 0, 3, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 3, 3, 0]

•
•
•

코드 설명

```
import sys
from collections import deque
```

```
N, K = map(int, sys.stdin.readline().split())
```

 입력

```
q = deque()
```

 BFS 사용 위한 queue

```
q.append(N)
```

 시작위치 삽입

```
visited = [0 for _ in range(100001)]
```

 해당 인덱스 위치까지 걸리는 이동 횟수
저장 배열 초기화

```
def bfs():
```

```
    while q:
```

```
        x = q.popleft()
```

```
        if x == K:
```

 K : 도착 위치

```
            return visited[x]
```

```
        for i in [x-1, x+1, 2*x]:
```

 이동할 수 있는 부분 확인

```
            if 0 <= i <= 100000 and visited[i] == 0:
```

 범위 내, 방문 하지 않은곳

```
                visited[i] = visited[x] + 1
```

 이동횟수 저장

```
                q.append(i)
```

 삽입

```
print(bfs())
```

감사합니다