

boj 6087 레이저 통신

- 문제: 'C' → 'C'로 가는데 설치해야 할 거울 수 최소화

· 거울 설치하는 '.' 이면 어디든 가능.

· '*'은 벽. (지나갈 수 없다)

- 접근.

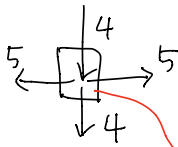
· 큐에 들어갈 정보는 네 가지가 필요하다.

(
1. 다음 y값
2. 다음 x값
3. 방향 (동서남북) dir
4. 다음 좌표까지 달는데 필요한 거울 수.
) → tuple로 구현.

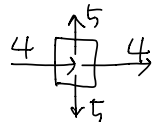
· 현재까지 거울 최소값을 담은 int형 배열이 필요하다.

· 각 방향에 대한 방문 체크는 필요 없다. why?

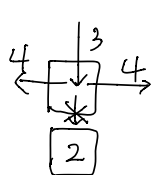
ex)



원래 4였다면 꺾이는 (거울 설치해야 하는) 경로는 continue;
↳ 원래 4라는 말은 이미



← 이것이 지나갔다는 말



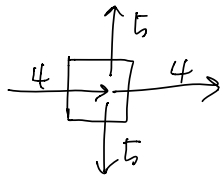
또는
← 이것이 지나갔다는 말.

두 경우 모두 꺾여서 5가 되는 경로가 지나갈 필요가 없다.

but, 원래 4였더라도 직진 경로는 continue;
or

꺾이는 경로더라도 같은 값.
(+한 결과)

반례)



← 이 경로에 의해 이미 4로 (같은 값으로)
갱신되어 있더라도 다른 방향 (\rightarrow_4 or \downarrow_4)
에 대해서는 지나가야 할 필요가 있음.

⇒ 결국 꺾이든 직진하든 상관없이
(cnt+1) (cnt)

방향에 따라 $(new_cnt = dir == ? ? cnt : cnt+1)$ 을

처리해준 뒤에 new_cnt가 더 작거나 같다면 다음 좌표값을
갱신해주고 pass 시켜야 할 필요가 있다.

* 네 방향별로 최소값을 갖는 visited를 체크한다면
이 문제에서는 100×100 이기 때문에 무조건 시간초과.
모든 좌표가 '.' 이라면 $O(4^n)$ 시간이 걸린다. (추정).

* 모든 방향에 대해 현재까지의 거울 수를 비교하기만 한다면
어느 방향으로 가는 경로든지 상관없이 모든 경우를 확인해볼 수 있다.
'.' 가면 같은 조건 (만났을 때 확인할 수 있는 경우가 분명함).
(모든 '.' 에는 거울을 설치하거나 설치하지 않을 수 있음).

boj 2151 거울 설치

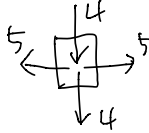
- 문제: 레이저 통신 문제와 거의 똑같아 보이는 문제.

bwt, 거울을 설치할 수 있는 '.' 이

< 적절만 가능한 '.' 가 따로 주어진다.

이 많은 위의 문제 처럼

'!'에서)



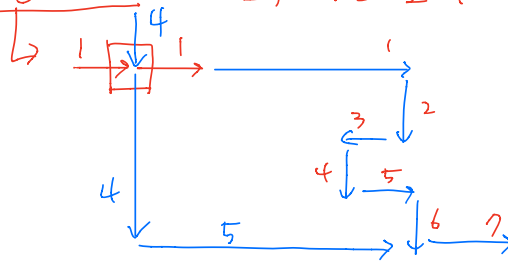
← 방향 관계없이 현재까지 필요한 거울 수만 비교하면 됨.

하지만,

'.'에서)



기준에 1이더라도, 4를 보내야 한다.



↑
이런 경우가 생기기 때문에.

* 하지만 '.' 이더라도 방향이 같을 때는 거울 수 체크를 안할 수 있다.

⇒ 결국 방향별로 거울 수를 비교해야 한다는 말인데..

그래서 처음에는, 방향별 방문 배열 체크하는 1111 비트 사용하는 배열
+ 각 좌표의 최소값 저장하는 2차원 배열로 구현했지만

⇒ 같은 경로에 의해서가 아닌 독립된 경로에 대해

2가지의 배열이 각각 생성되어 최적 경로가

마지막에 방문 체크했을 때 (해당 방향도 0 이며
최소 거울 수도 0 이)

지나갈 수 없는 문제가 발생.

⇒ 결국 좌표에 따른 방향 값이 종속적으로 관리되도록
 $[y][x][0\sim3]$ 의 3차원 배열을 사용하여 해결.

↳ 이 문제는 레이저 통신과 다르게

50X50이고 문의위(지도 테두리)에만 존재.

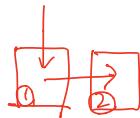
시간 초과 X였지만 계산은 쉽지 않음...

*여기서 방향은

동 - 0
서 - 1
남 - 2
북 - 3

이며, 기점은 직선 레이저의 진행 방향이다.

(여기서 앞는지가 아니라 해당 노드에 대한 진행 방향.)



: 이렇게 진행되어 왔다면 0.

①에서 꺾여서 왔는지 말든지 상관없이
 $visited[y][x][i]$ 를 갱신했다면
그 노드를 기준으로 한 방향을 말함.

(이렇게 되면 해당 방향은 다른 방향의
거울 수에 독립되기 때문에 거울 수가
같다면 continue해도 된다.)