village

	index	
	[0]	[ ]
1번도시	[1]	[2,3]
2번 "	[2]	[3,4]
3번 "	[3]	[ ]
4번 "	[4]	[ ]

went

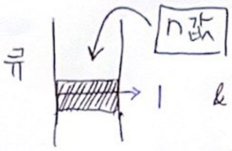
index		
[0]	-1	
[1]	-1	1번도시
[2]	-1	2번 "
[3]	-1	3번 "
[4]	-1	4번 "

x=1 //초기도시 x=1

find(x)

def find(n) <find 함수>

① 큐에 n값 넣는다



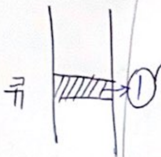
& went[1] = 0

초기 위치 나타냄

('0'번 이동했을때 얻어짐) 1번도시가

초기위치에

② while 큐가 비지 않은동안 (반복문)



큐에서 1 꺼냄

↳ vNum에 넣음

village[vNum] => village[1] => [2,3]

i = 2, 3

①

if went[2]==-1 한 번도 가지 x

참

went[2] = went[vNum] + 1

= went[1] + 1

= 0 + 1

∴ went[2] = 1 (1번 이동하면 2번도시로 갈수 있음)

초기 위치에서

⊕ 큐 2

②

if went[3]==-1 한 번도 가지 X

참

went[3] = went[vNum] + 1

= went[1] + 1

= 0 + 1

∴ went[3] = 1 (1번 이동하면 3번도시로 갈수 있음)

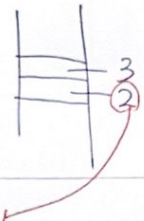
초기 위치에서

⊕ 큐 3 2

③ while queue

while loop 한 번 돈 후, 큐가 비었는지 check

현재 큐



이므로 비지 X

현재 went

index	
[0]	-1
[1]	0
[2]	1
[3]	1
[4]	-1

① 큐에서 2 꺼냄 → vNum = 2

② village[vNum] ⇒ village[2] = [3, 4]

i = 3, i = 4

(1) (2)

if went[3]==-1 거짓 3도시에 한번도 가지 X

참이었다면
went[3] = went[vNum] + 1
= went[2] + 1
= 1 + 1

∴ went[3] = 2

(초기 위치에서 2번 이동하면 3번도시로 갈수 있음)

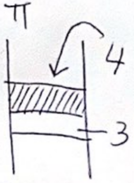
if went[4] == -1 4도시에 한번도 가지 X

참 went[4] = went[vNum] + 1
= went[2] + 1
= 1 + 1

∴ went[4] = 2

(초기위치에서 2번 이동하면 4번도시로 갈 수 있음)

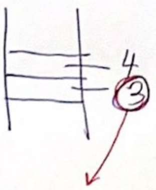
④ 큐



④ while queue

while루프 돈 후, 큐가 비었나 check

현재 큐



이므로 비지 X

현재 went

index	
[0]	-1
[1]	0
[2]	1
[3]	1
[4]	2

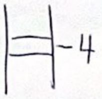
① 큐에서 3 꺼냄 → vNum = 3

② village[vNum] → village[3] = [ ]

for문 돌지 X

⑤ while queue

현재 큐



이므로 비지 X

← 현재 went

① 큐에서 4 꺼냄 → vNum = 4

② village[vNum] → village[4] = [ ]

for문 돌지 X

⑥ while queue ← 현재 큐 | | 비었음!

→ while loop 빠져나옴

→ find 함수 빠져나옴

```
              마을수
                ↓
for i in range(n+1):   i=0 ~ n
    if went[i] == k:
        print(i) → 인덱스 값(마을 번호) 출력
if k not in went:   k가 went에 없으면
    print(-1)   ← -1 출력
```

현재 went		
index		
[0]	-1	X
[1]	0	마을 1
[2]	1	마을 2
[3]	1	마을 3
[4]	2	마을 4