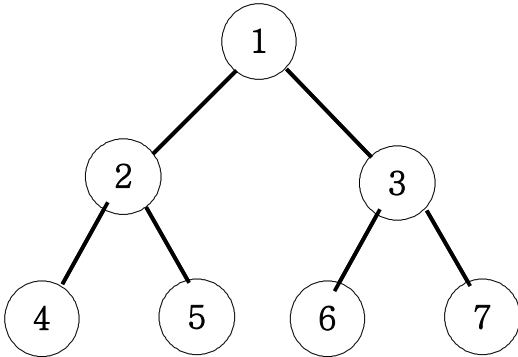


## 이진트리 레벨탐색(너비우선탐색 : BFS)

아래 그림과 같은 이진트리를 큐(Queue) 자료구조를 이용해 레벨탐색을 해보세요.



레벨탐색 출력 : 1 2 3 4 5 6 7

```
from collections import deque
def BFS():
    Q = deque()
    Q.append(1)
    L = 0
    while Q:
        n = len(Q)
        print(L, end = " : ")
        for i in range(n):
            v = Q.popleft()
            print(v, end = " ")
            for nv in [v*2, v*2+1]:
                if nv > 7:
                    continue
                Q.append(nv)
        print()
        L += 1
```

BFS()

## 최소 점프

현수는 놀이터에서 놀다가 집으로 가려고 합니다. 놀이터의 위치와 집의 위치가 수직선상의 좌표 점으로 주어집니다. 놀이터는 수직선상의 0지점입니다.

현수는 놀이터에서 스카이콩콩을 타고 점프를 하면서 집으로 이동하려고 합니다.

점프는 다음과 같은 규칙으로 합니다.

- 1) 현재 지점에서 앞으로 +1 만큼 점프이동할 수 있습니다.
- 2) 현재 지점에서 뒤쪽으로 -1 만큼 점프이동할 수 있습니다.
- 3) 현재 지점에서 앞쪽으로 +5 만큼 긴 점프이동을 할 수 있습니다.

매개변수 home에 현수의 집의 위치가 주어지면 놀이터에서 집까지 최소 몇 번의 점프로 집에 도착할 수 있는지 최소 점프횟수를 구하여 반환하세요.

입출력 예:

home	answer
10	2
14	4
25	5
24	6
345	69

제한사항:

- 수직선의 좌표는 0부터 10,000까지입니다.
- 현수가 집으로 반드시 갈 수 있습니다.

## 검정색 영역구하기

5 \* 5 이차원 배열에 모니터 화면을 표현합니다. 모니터의 화면은 최초 검정색과 흰색으로만 표현되어 있습니다. 검정색은 1, 흰색은 0으로 표현됩니다.

	0	1	2	3	4
0	0	1	1	0	0
1	0	1	1	0	0
2	0	1	0	0	0
3	0	0	0	1	0
4	0	0	1	1	0

상하좌우로 1(검정색)이 연결되어 있으면 한 영역으로 간주합니다.

화면의 격자 정보가 위와 같다면 검정색으로 칠해진 영역은 2곳입니다.

매개변수 board에 모니터 화면의 격자정보가 주어지면 검정색으로 칠해진 영역은 총 몇 개가 있는지 구하여 반환하는 프로그램을 작성하세요.

입출력 예:

board	answer
[[0, 1, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	2
[[1, 1, 1, 0, 1], [1, 1, 1, 0, 1], [0, 0, 1, 0, 0], [1, 1, 0, 1, 0], [1, 0, 1, 0, 0]]	5
[[0, 0, 1, 0, 0], [0, 1, 1, 0, 0], [0, 1, 0, 0, 0], [1, 0, 0, 1, 0], [0, 0, 1, 1, 0]]	3
[[0, 0, 0, 0, 1], [0, 0, 1, 0, 0], [0, 1, 0, 1, 0], [0, 0, 0, 1, 0], [0, 0, 1, 0, 0]]	5

제한사항:

- 검정색 영역은 1개 이상 반드시 존재합니다.