Editorial :
process can be simulated there will be no more than 35 positions a knight can attack in two moves . Run a DFS upto depth of 2 or use iterative approach to list all the positions reachable .

NOTE:
All the position attackable in 2 moves can also precomputed for each Position but for our problem that was not required .

Complexity :
O(1)

CODE :

```
for _ in range(int(input())):
    a,b,c,d=map(int,input().split())

    #listing all the moves of a knight
    delta=[(2,1),(2,-1),(-2,-1),(-2,1),(1,2),(-1,2),(1,-2),
(-1,-2)]
    attack_position=set()
    first_attack=[]
    #generating all the position knight can attack in one move
    for dx,dy in delta :
        if 1<=a+dx<=8 and 1<=b+dy<=8:
            attack_position.add((a+dx,b+dy))
            first_attack.append((a+dx,b+dy))

    #generating all the position knight can attack in two move
    for dx,dy in delta :
        for aa,bb in first_attack:
            if (1<=aa+dx<=8 and 1<=bb+dy<=8):
                attack_position.add((aa+dx,bb+dy))

    #if the second knight is at any of the attack position
    if (c,d) in attack_position :
        print('YES')
    else:
        print('NO')
```