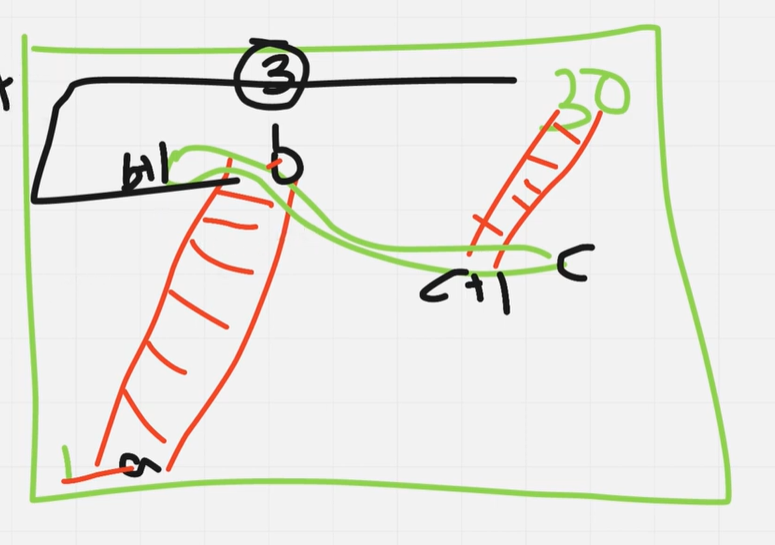


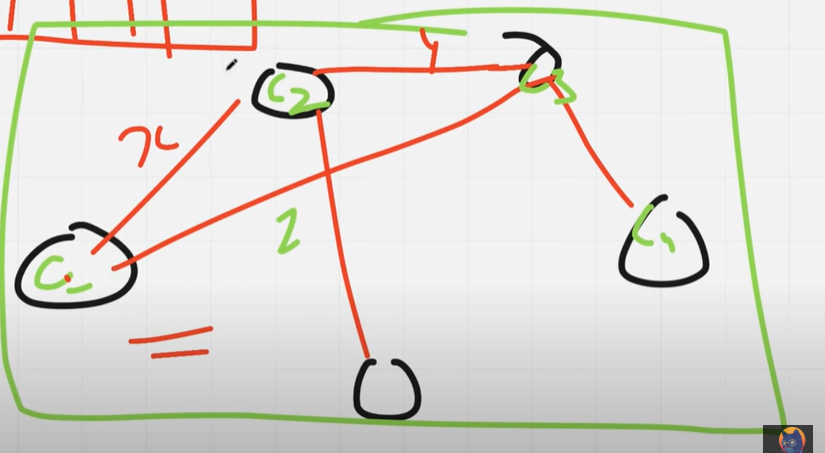
Logic:

See sometime snake cutting you is beneficial too.



See this example see we move left to right and then right to left in out game.so above snake there is beneficial as it will take me to the 30 directly.

Here we will use BFS.



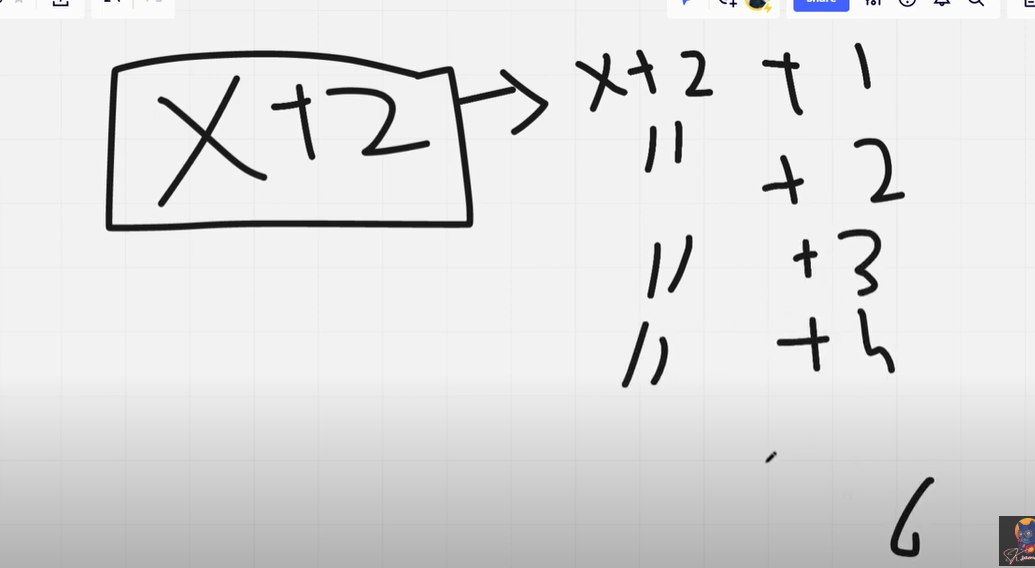
See graph is used when there is connection in between.

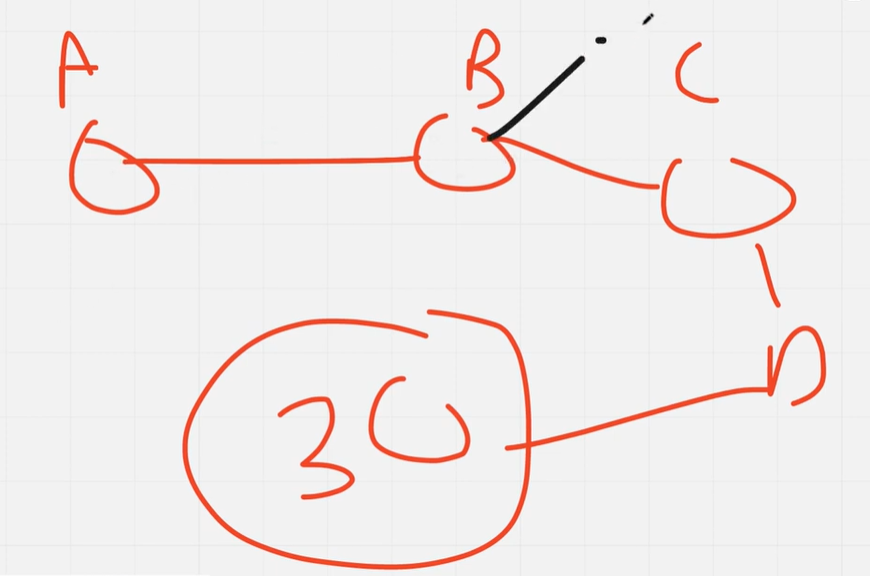
A picture containing text, whiteboard

Description automatically generated

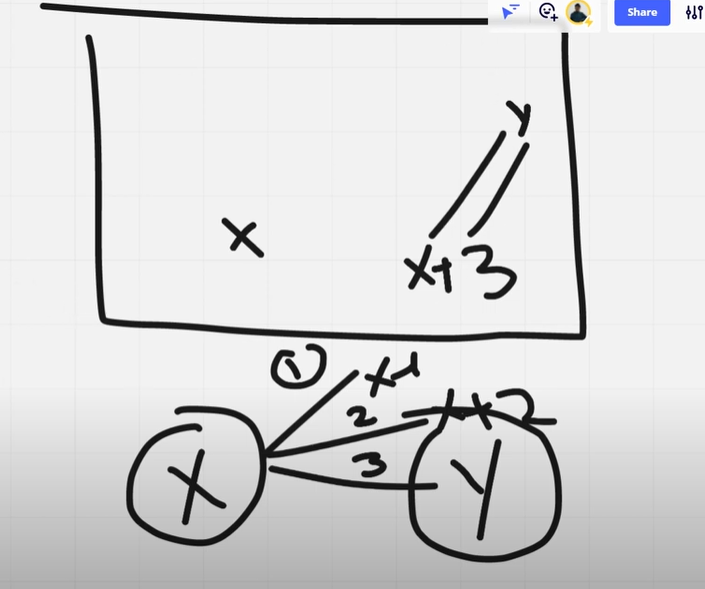
See dice can take me from x+1 to x+6 so we will check where we can go to.

Consider we went to x+2 cell then from there we can go to .





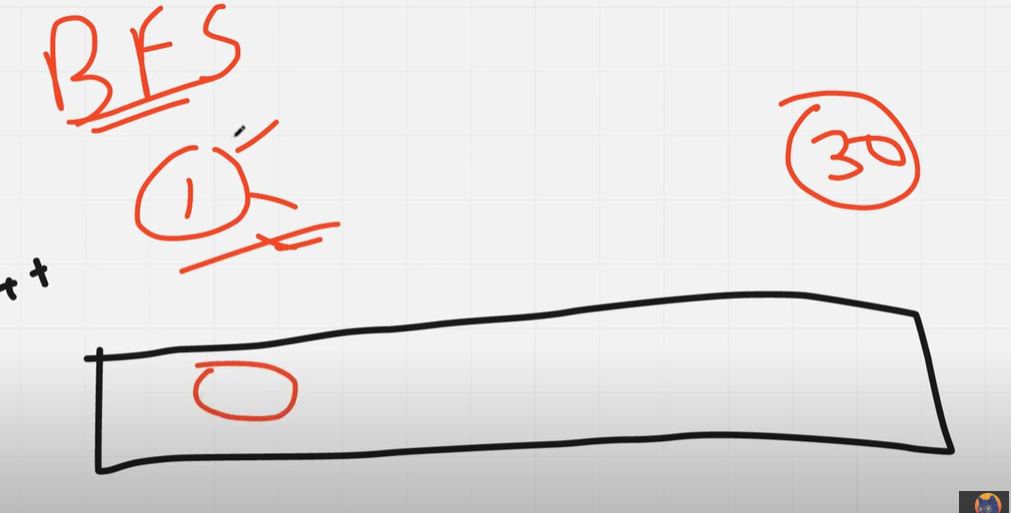
See many route will be formed that will tell us that all the path to 30.we will use bfs to solve this issue.



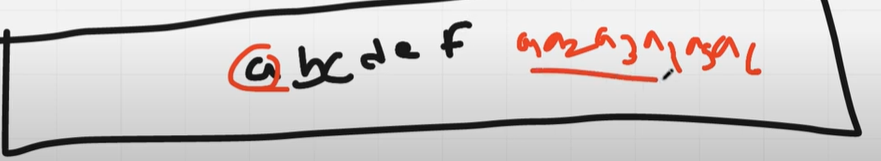
See above game showing that x+3 we move to y. Hence, we see that x to x+3 will not be shown but y.

This was the main logic. now will be shown the implementation.

See we will use a queue for this.

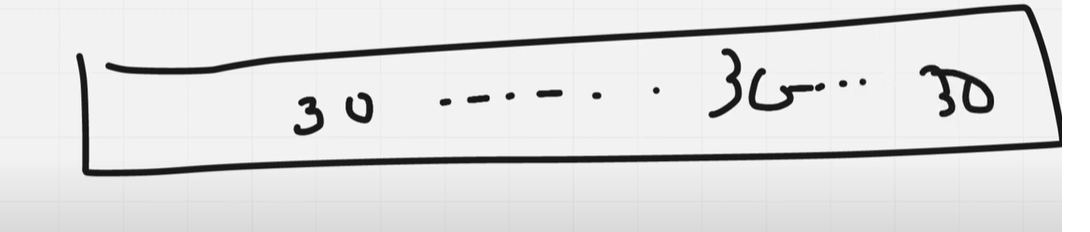


See we will first take 1 cell and put this in the queue.

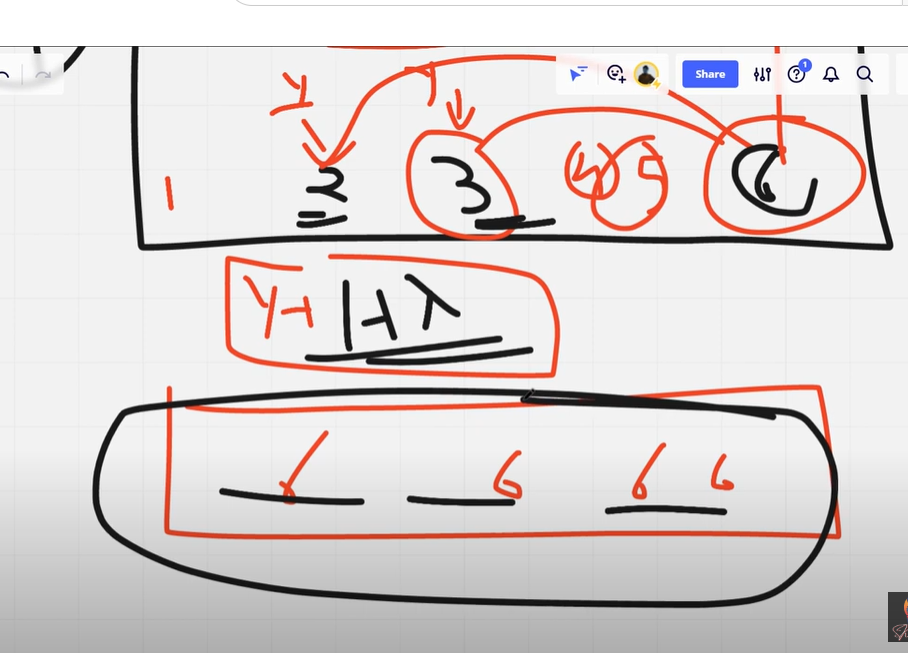


See here a is put in the queue then we see a+1 to a+6 as a1….a6.same for b as b1 …b6 .same for cdef.

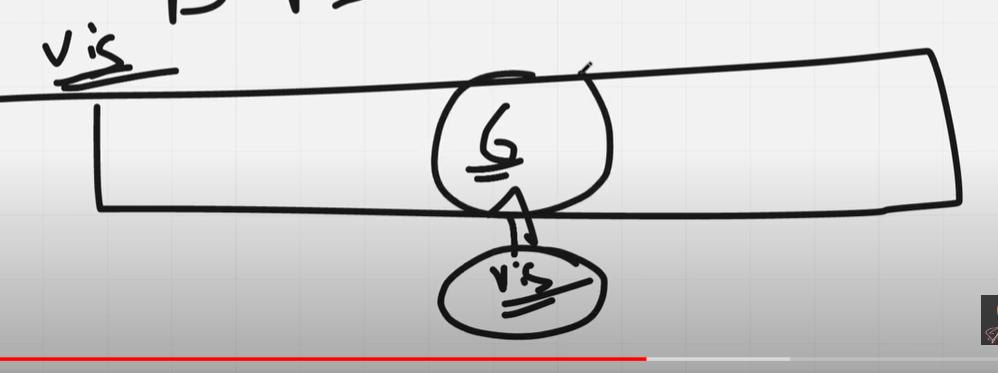
Will put the same in the queue. See this a1 shows the number like 2(1 is home as a then 1 comes this will go to 2 in board).



Also see 30 will come many times as many route will be there.

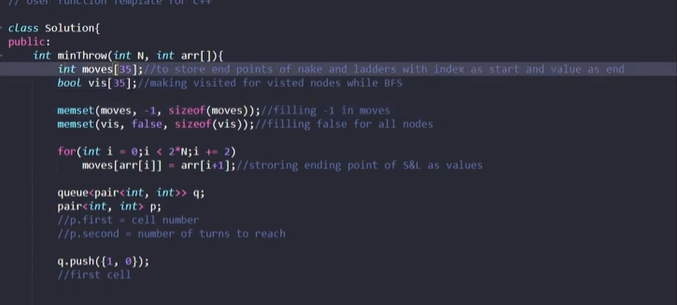


See here 1 thing will take extra time consider that 2,3,4,5 has to go to 6 and from 6 ladder to 30.hence 6 will come many time .so rather than every time visiting we will be using a visited array to solve this issue.

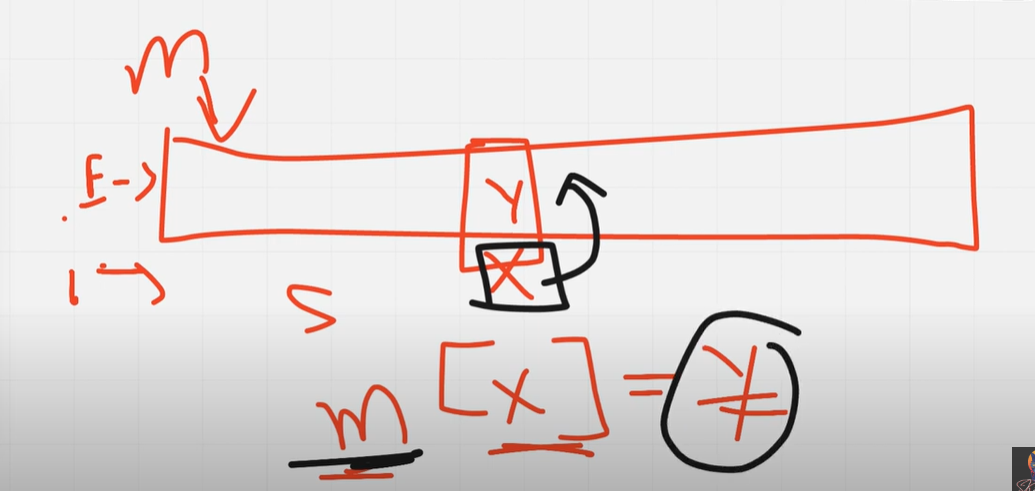


We will check in visited array and then if confirmed make sure we do not put this in the queue.

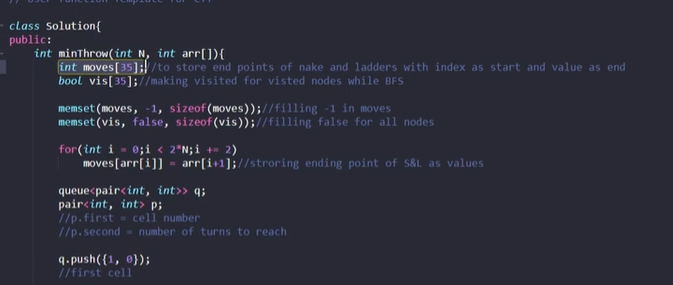
Code:



Moves [35] is

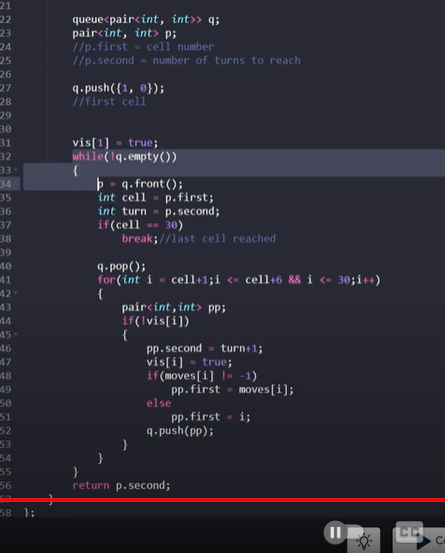


See this index is x the starting point and value stored is y which tell us the ending point. Meaning x take us to the m[x].





Because we need to know the number of moves needed to reach that cell.



 see bfs hence next step +1 will be happening simultaneously.