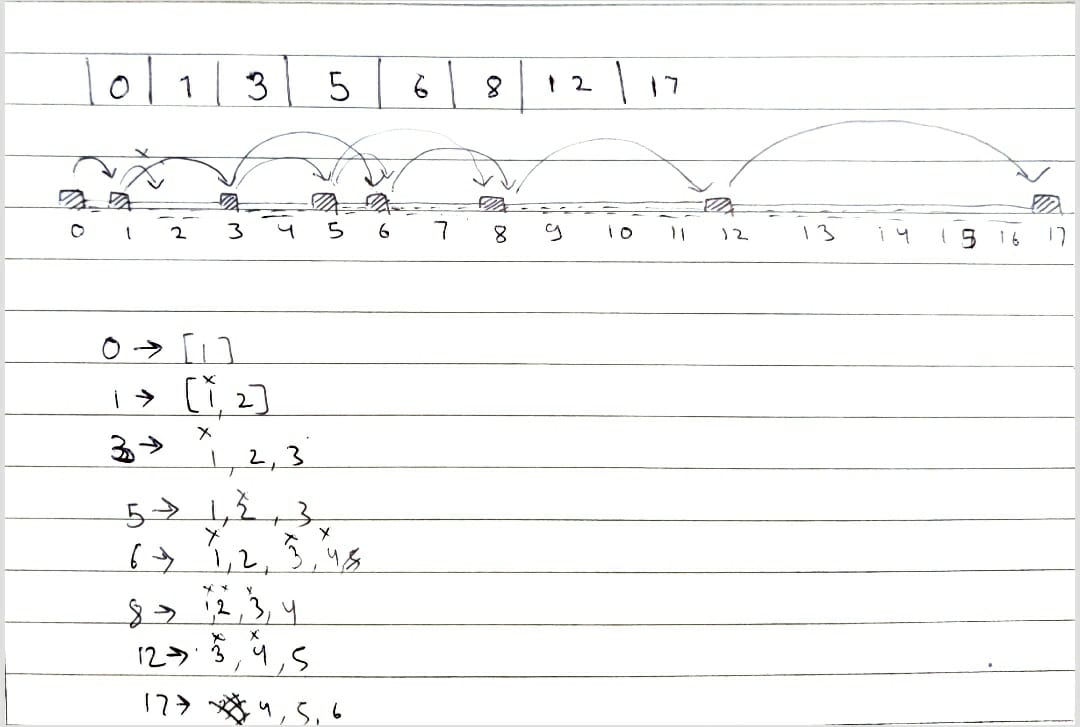
Question: https://leetcode.com/problems/frog-jump/

So we store array elements in a hashmap with empty sets. Now initially we add 1 to the stones[0] key’s set.

Now start iterating through the stones array.

For ith element, We check its set in map and traverse through the set,and check if ar[i] + jump exists in map or not, if it exists then add jump-1, jump, jump+1 to the set of ar[i]+jump key.



Code:  
class Solution {

public boolean canCross(int[] stones) {

HashMap<Integer, Set<Integer>> jumpRecord = new HashMap<>();

for(int stone: stones) jumpRecord.put(stone, new HashSet<Integer>());

Set<Integer> temp;

temp=jumpRecord.get(stones[0]);

temp.add(1);

jumpRecord.put(stones[0], temp);

for(int stone: stones){

for(int jump:jumpRecord.get(stone)){

if(jumpRecord.containsKey(stone+jump)){

if(stone+jump==stones[stones.length-1]){

return true;

}

if(jump==1){

temp=jumpRecord.get(stone+jump);

temp.add(1);

temp.add(2);

jumpRecord.put(stone, temp);

}

else{

temp=jumpRecord.get(stone+jump);

temp.add(jump-1);

temp.add(jump);

temp.add(jump+1);

jumpRecord.put(stone, temp);

}

}

}

}

return false;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>