Question: https://leetcode.com/problems/longest-arithmetic-subsequence/submissions/

So the idea here is to find the longest sub-sequence of AP.

Thus what we do is we maintain an array of HashMap, each map stores diff as key and length of sub-sequence as value till ith position,

Now since we are looking for previous positions we check if that diff has occurred for that position’s hashmap or not, if then it will add up to give the sub-sequence.

So we go on checking for max sub-sequence while checking all differences.

Code:  
class Solution {

public int longestArithSeqLength(int[] nums) {

int dp[][]=new int[nums.length][nums.length];

HashMap<Integer, Integer> mem[] = new HashMap[nums.length];

int res=Integer.MIN\_VALUE;

for(int i=0; i<nums.length; i++){

mem[i]= new HashMap<>();

int base=nums[i];

for(int j=0; j<i; j++){

int diff=nums[j]-base;

mem[i].put(diff, Math.max(mem[j].getOrDefault(diff,0), mem[j].getOrDefault(diff,0)+1));

res=Math.max(res,mem[i].get(diff));

}

}

return res+1;

}

}

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