

DAY 50

INTERVIEW BIT PROBLEMS :

1. 4 Sum

Given an array S of n integers, are there elements a , b , c , and d in S such that $a + b + c + d = \text{target}$? Find all unique quadruplets in the array which gives the sum of target.

Note:

Elements in a quadruplet (a,b,c,d) must be in non-descending order. (ie, $a \leq b \leq c \leq d$)

The solution set must not contain duplicate quadruplets.

Example :

Given array $S = \{1\ 0\ -1\ 0\ -2\ 2\}$, and target = 0

A solution set is:

$(-2, -1, 1, 2)$

$(-2, 0, 0, 2)$

$(-1, 0, 0, 1)$

Also make sure that the solution set is lexicographically sorted.

$\text{Solution}[i] < \text{Solution}[j]$ iff $\text{Solution}[i][0] < \text{Solution}[j][0]$ OR $(\text{Solution}[i][0] == \text{Solution}[j][0] \text{ AND } \dots \text{Solution}[i][k] < \text{Solution}[j][k])$

CODE :

PYTHON

class Solution:

 # @param A : list of integers

 # @param B : integer

 # @return a list of list of integers

 def fourSum(self, A, B):

 res = set()

 store = dict()

 if not A or len(A)==0:

 return list(res)

 A.sort()

 for i in range(len(A)):

 for j in range(i+1, len(A)):

 curSum = A[i]+A[j]

 if (B-curSum) in store:

 for f, s in store[B-curSum]:

 if i>s:

 res.add((A[f], A[s], A[i], A[j]))

 if (curSum) not in store:

 store[curSum] = [(i, j)]

 else:

 store[curSum].append((i, j))

 res = list(res)

 res.sort()

 return res

JAVA

```
public class Solution {
    public ArrayList<ArrayList<Integer>> fourSum(ArrayList<Integer> a, int b) {
        Collections.sort(a);
        ArrayList<ArrayList<Integer>> out= new ArrayList<ArrayList<Integer>>();
        for (int i=0; i<a.size()-3; i++){
            for (int j=i+1; j<a.size()-2; j++){
                int tot= b-a.get(i)-a.get(j);
                int k=j+1;
                int l=a.size()-1;
                while (k<l){
                    if (a.get(k)+a.get(l)>tot) l--;
                    else if (a.get(k)+a.get(l)<tot) k++;
                    else{
                        ArrayList<Integer> temp= new ArrayList<Integer>();
                        temp.add(a.get(i)); temp.add(a.get(j));
                        temp.add(a.get(k)); temp.add(a.get(l));
                        if (!out.contains(temp)) out.add(temp);
                        k++; l--;
                    }
                }
            }
        }
        return out;
    }
}
```