DAY 33

INTERVIEW BIT PROBLEMS :

1. Subset

Given a set of distinct integers, S, return all possible subsets.

Note:
Elements in a subset must be in non-descending order.
The solution set must **not contain duplicate subsets.**
Also, the subsets should be **sorted in ascending** (lexicographic) order.
The list is not necessarily sorted.
Example :
If S = [1,2,3], a solution is:
[
[],
[1],
[1, 2],
[1, 2, 3],
[1, 3],
[2],
[2, 3],
[3],
]

CODE :

PYTHON

```
class Solution:
    # @param A : list of integers
    # @return a list of list of integers
    def subsets(self, A):
        #reversing the input list
        A.sort(reverse=True)
        n=len(A)
        #initializing the output list
        out=[]
        for i in range(n):
            ele=[A[i]]
            temp=[ele+x for x in out]
            if len(out)!=0:
                out+=temp
            out+=[ele]
        out.append([])
        out.reverse()
        return out
```

C++

```
void subset(vector<int> &A,vector<vector<int>> &ans,vector<int> temp,int index)
{
      ans.push_back(temp);

      for(int i=index;i<A.size();i++)
      {
            temp.push_back(A[i]);
            subset(A,ans,temp,i+1);
            temp.pop_back();
      }
      return;
}

vector<vector<int> > Solution::subsets(vector<int> &A) {
      vector<vector<int>> out;
      sort(A.begin(),A.end());
      int index=0;
      vector<int> temp;
      subset(A,out,temp,index);
   return out;
```