491. Increasing Subsequences

Add to List

Description | Submission | Solutions

- Total Accepted: **4582**
- Total Submissions: **11742**
- Difficulty: **Medium**
- Contributors: **Stomach_ache**

Given an integer array, your task is to find all the different possible increasing subsequences of the given array, and the length of an increasing subsequence should be at least 2 .

Example:

```
Input: [4, 6, 7, 7]

Output: [[4, 6], [4, 7], [4, 6, 7], [4, 6, 7, 7], [6, 7], [6, 7, 7], [7,7], [4,7,7]]
```

Note:

1. The length of the given array will not exceed 15.
2. The range of integer in the given array is [-100,100].
3. The given array may contain duplicates, and two equal integers should also be considered as a special case of increasing sequence.

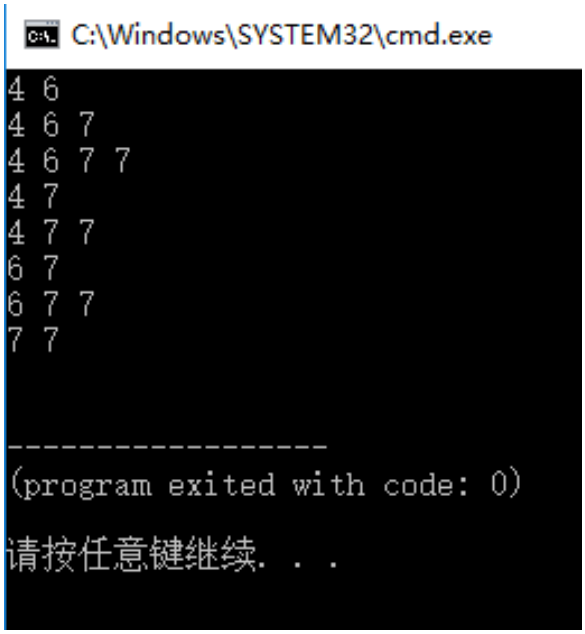
Subscribe to see which companies asked this question.

```
#include<iostream>
#include<stdio.h>
#include<algorithm>
#include<unordered_set>
#include<string>
using namespace std;
// cancel continuous chars with nums >= 3
```

```
void dfs(vector<vector<int>>& res, vector<int>& seq, vector<int>& nums, int pos)
{
    if(seq.size()>1) res.push_back(seq);
    unordered_set<int> hash;
    for(int i=pos;i<(int)nums.size();i++)
    {
        if((seq.empty() || nums[i]>=seq.back()) && hash.find(nums[i])==hash.end())
        {
            seq.push_back(nums[i]);
            dfs(res,seq,nums,i+1);
            seq.pop_back();
            hash.insert(nums[i]);
        }
    }
}

vector<vector<int>> findSubsequences(vector<int>& nums)
{
    vector<vector<int>> res;
    vector<int> seq;
    dfs(res, seq, nums, 0);
    return res;
}

int main(int argc,char *argv[])
{
    vector<int> nums = {4,6,7,7};
    vector<vector<int>> res = findSubsequences(nums);
    for(int i=0;i<(int)res.size();++i)
    {
        for(int j=0;j<(int)res[i].size();++j) printf("%d ",res[i][j]);
        printf("\n");
    }
    return 0;
}
```


C:\Windows\SYSTEM32\cmd.exe
4 6
4 6 7
4 6 7 7
4 7
4 7 7
6 7
6 7 7
7 7

(program exited with code: 0)
请按任意键继续. . .