

## 368. Largest Divisible Subset

题目描述: <https://leetcode.com/problems/largest-divisible-subset/>

给定一个数组，求这个数组满足如下要求的最长子串。  
子串中所有的元素都可以互相整除。

解题思路:

dp

如果一个数可以和这个集合的最小公倍数整除则就可以整除所有元素。

注意要先排序

代码:

```
class Solution {
public:
    int findZXGBS(int a, int b) {
        int tmpa = a, tmpb = b;
        int r = 1;
        while(b > 0) {
            r = a % b;
            a = b;
            b = r;
        }
        return tmpa/a*tmpb;
    }
    vector<int> largestDivisibleSubset(vector<int>& nums) {
        //整数最小公倍数。
        int len = nums.size();
        if(len == 0) {
            return {};
        }
        sort(nums.begin(), nums.end());
        vector<int> zxgbs(len, 0);
        vector<int> f(len, 1);
        vector<int> loc(len, -1);
        vector<int> resv;
        for(int i = 0; i < len; i++) {
            zxgbs[i] = nums[i];
        }
        for(int i = 0; i < len; i++) {
            for(int j = 0; j < i; j++) {
```

```

        if(nums[i] % zxgbs[j] == 0) {
            f[i] = max(f[j] + 1, f[i]);
            if(f[i] == f[j] + 1) {
                loc[i] = j;
                if(zxgbs[j] > nums[i]) {
                    zxgbs[i] = findZXGBS(zxgbs[j], nums[i]);
                }
                else {
                    zxgbs[i] = findZXGBS(nums[i], zxgbs[j]);
                }
            }
        }
    }
}

int res = f[0];
int k = 0;
for(int i = 1; i < len; i++) {
    res = max(res, f[i]);
    if(res == f[i]) {
        k = i;
    }
}

for(int i = k; i >= 0; i--) {
    resv.push_back(nums[i]);
    i = loc[i];
}

return resv;
}

};

```