Largest divisible pairs subset

```
#include <bits/stdc++.h>
using namespace std;

// 1 3 6 13 17 18
// dp[i]: arr[i]作为最小元素时，该子集中元素的个数

int largestSubset(vector<int>& arr, int N) {
    vector<int> dp(N, 1);
    int maxSubset = INT_MIN;
    sort(arr.begin(), arr.end());
    for (int i = 0; i < N; ++i) {
        int pre = arr[i];
        for (int j = i + 1; j < N; ++j) {
//            printf("arr[%d] = %d, pre = %d", j, arr[j], pre);
            if (arr[j] % pre == 0) {
                dp[i]++;
                pre = arr[j];
            }
//            printf(", dp[i] = %d\n", dp[i]);
        }
        maxSubset = max(maxSubset, dp[i]);
    }
    return maxSubset;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        vector<int> arr;
        scanf("%d", &N);
        for (int i = 0; i < N; ++i) {
            int n;
            scanf("%d", &n);
            arr.push_back(n);
        }
        printf("%d\n", largestSubset(arr, N));
    }
    return 0;
}
```