N meetings in one room

按照endTime进行排序，只要后一个的startTime ≤ 上一次会议的endTime，就选择这个会议作为下一次会议

```
#include <stdio.h>
#include <vector>
#include <bits/stdc++.h>

// sort meetings by end time
// meeting <idx, <startTime, endTime>>
void sortMeetings(std::vector<std::pair<int, std::pair<int, int>>> &meetings) {
    std::sort(meetings.begin(), meetings.end(),
              [](const std::pair<int, std::pair<int, int>> &a, const std::pair<int, std::pair<int, int>> &b) {
                  return a.second.second < b.second.second;
              });
    printf("%d", meetings[0].first);
    int lastEndIdx = 0;
    for (int i = 1; i < meetings.size(); ++i) {
        if (meetings[lastEndIdx].second.second <= meetings[i].second.first) {
            printf(" %d", meetings[i].first);
            lastEndIdx = i;
        }
    }
    printf("\n");

}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        std::vector<int> start;
        std::vector<int> end;
        for (int i = 0; i < N; ++i) {
            int startTime;
            scanf("%d", &startTime);
            start.push_back(startTime);
        }
        for (int i = 0; i < N; ++i) {
            int endTime;
            scanf("%d", &endTime);
            end.push_back(endTime);
        }
        std::vector<std::pair<int, std::pair<int, int>>> meetings;
        for (int i = 0; i < N; ++i) {
            meetings.push_back(std::make_pair(i + 1, std::make_pair(start[i], end[i])));
        }
        sortMeetings(meetings);
    }
    return 0;
}
```