```
typedef pair<int,int> pii;
static bool cmp(vector<int> & a,vector<int> & b){
    if(a[0] == b[0]) return a[1] < b[1];
    return a[0] < b[0];
}

int AllTime(vector<vector<int>> & task){
    int res = 0;
    int wait = 0;
    int curr = 0;
    int mod = 1e9+7;
    priority_queue<int,vector<int>,greater<int>> pq;
    sort(task.begin(),task.end(),cmp);

    curr = task[0][0];
    for(int i = 0; i < task.size(); ++i){
        if(pq.empty() || task[i][0] <= curr){
            pq.push(task[i][1]);
            continue;
        }

        if(task[i][0] > curr){
            while(!pq.empty() && curr + pq.top() <= task[i][0]){
                res = (res + pq.top()*(pq.size()-1))%mod;
                curr = curr + pq.top();
                pq.pop();
            }
            if(!pq.empty()){
                wait = task[i][0] - curr;
                int rest = curr + pq.top() - task[i][0];
                res = (res + wait*(pq.size()-1))%mod;
                pq.pop();
                pq.push(rest);
            }
            curr = task[i][0];
            pq.push(task[i][1]);
        }
    }

    while(!pq.empty()){
        res = (res + pq.top()*(pq.size()-1))%mod;
        pq.pop();
    }

    return res;
}

int main(){
    vector<vector<int>> task;
    task.push_back({0,5});
    task.push_back({1,2});
    //task.push_back({1,2});
    task.push_back({2,1});
    cout<<AllTime(task);
}
```