Rent your Airplane and make Money

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
using namespace std;

struct Task
{
    int st, dur, price;

    bool operator<(const Task &t) const
    {
        return st < t.st;
    }
    bool operator<(const int a) const
    {
        return st < a;
    }
};
int tasksPicker(vector<Task> &vt)
{
    vector<int> prices(vt.size()+1);
    prices[vt.size()-1] = vt[vt.size()-1].price;
    for (long i = vt.size()-2; i >= 0 ; i--)
    {
        long j=lower_bound(vt.begin()+i+1,vt.end(),vt[i].st+vt[i].dur)-vt.begin();
        prices[i] = max(prices[i+1], vt[i].price+prices[j]);
        cout<<vt[i].st+vt[i].dur<<' '<<j<<' '<<prices[i]<<endl;
    }
    for (int i=0; i<prices.size(); i++) {
        cout<<prices[i]<<endl;
    }
    return prices[0];
}
void RentyourAirplaneandmakeMoney()
{
    int T = 0, n = 0;
    cin>>T;
    while (T--)
    {
        cin>>n;
        vector<Task> vt(n);
        for (int i = 0; i < n; i++)
        {
            cin>>vt[i].st>>vt[i].dur>>vt[i].price;
        }
        sort(vt.begin(), vt.end());
        cout<<tasksPicker(vt)<<endl;
    }
}

int main()
{
    RentyourAirplaneandmakeMoney();
    return 0;
}
```