

**P1833 櫻花**

### 题目描述：

lester大神后院里种了 $n$ 棵樱花树，每棵都有美学值 $C_i$ 。lester大神在每天上学前都会来赏花。lester大神可是生物学霸，他懂得如何欣赏樱花：一种樱花树看一遍过，一种樱花树最多看 $A_i$ 遍，一种樱花树可以看无数遍。但是看每棵樱花树都有一定的时间 $T_i$ 。lester大神离去上学的时间只剩下一小会儿了。求解看哪几棵樱花树能使美学值最高且lester大神能准时（或提早）去上学

### 输入输出格式：

输入：第1行三个数：现在时间 $T_s$ （几点:几分），去上学的时间 $T_e$ （几点:几分）， $T_e - T_s \leq 200$ 分。lester大神院子里有几棵樱花树 $n$ （ $\leq 30$ ）

第2行~第 $n+1$ 行每行三个数：看完第 $i$ 棵树的耗费时间 $T_i$ ，第 $i$ 棵树的美学值 $C_i$ ，看第 $i$ 棵树的次数 $P_i$ （ $P_i=0$ 表示无数次， $P_i$ 是其他数字表示最多可看的次数 $P_i$ ）

输出：只有一个整数，表示最大美学值

### 样例输入：

```
6:50 7:00 3
2 1 0
3 3 1
4 5 4
```

### 样例输出：

```
11
```

# 建模

## → 显然是个背包问题

物品（花）可重复：只要看做多个相同物品即可  
可无限次重复的，实际有效次数为  $(T_e - T_s)/T_i$

## → 最优化目标是求和，有规约性

状态：  $f(i, t)$  表示  $t$  时间看前  $i$  棵最大价值

注意顺序不影响结果（无后效性）

决策：最后一棵树看几遍

转移方程：

$$f(i, t) = \max_{0 \leq k \leq P_i} \{f(i-1, t - kT_i) + kC_i\}$$

DP



# 解答

---

```
#define N 35
#define T 205
using namespace std;
int main() {
```

→ 有特殊输入格式时，scanf就很方便了

```
    int s1,s2,e1,e2,n=0;
    scanf("%d:%d %d:%d %d",&s1,&s2,&e1,&e2,&n);
    int time=e1*60+e2-s1*60-s2;//时间转换
    struct tree { // 结构体数组
        int t,c,p; // 时间, 价值, 次数
    }a[N];
```

→ 初始化结构体数组的简便写法

```
    for(int i=1;i<=n;i++){
        scanf("%d%d%d",&a[i].t,&a[i].c,&a[i].p);
        if(a[i].p==0){
            a[i].p=time/a[i].t;// 无限->有限
        }
    }
}
```

## 解答

---

```
int f[2][T]={};  
//背包问题  
for(int i=1;i<=n;i++){ // 阶段, 前i棵树  
    for(int t=0;t<=time;t++){ // 状态, 时间  
        f[i&1][t]=0;  
        for (int k=0;k<=a[i].p && t>=a[i].t*k;k++){ // 决策, 第i棵树看几遍  
            f[i&1][t]=max(f[i&1][t],f[1-i&1][t-a[i].t*k]+a[i].c*k);  
        }  
    }  
}  
printf("%d",f[n&1][time]);  
return 0;
```

- 复杂度:  $O(nT^2)$
- 使用了滚动数组节省空间