知识精炼（二）

主讲人：邓哲也

CF 864 E. Fire

现在一个房子里着火了，你需要从这个房间里抢救一些东西出来。

有 n 件物品，第 i 件物品的价值是 p[i]，抢救需要花费 t[i] 的时间，如果物品超过 d[i] 的时间还没有被救出来就会被烧掉。

问能救出的物品价值之和最大为多少，并且输出你能抢到的物品编号。

CF 864 E. Fire

典型的 01 背包问题。

这里的花费不是体积，而是时间。

用 f[i] 表示到 i 时刻最大能救多少价值。

$f[i] = \max(f[i], f[i-t[j]] + p[j])\ (i-t[j] \leqslant d[j])$

因此先对物品按 d[i] 从小到大排序，然后依次更新。

CF 864 E. Fire

```
struct node{
    int d, t, p, id;
}a[N];
int cmp(const node &a, const node &b){
    return a.d < b.d;
}
scanf(“%d”, &n);
for(int i = 1;i <= n;i ++)
    scanf(“%d%d%d”, &a[i].t, &a[i].d, &a[i].p);
sort(a + 1, a + n + 1, cmp);
```

CF 864 E. Fire

```
f[0] = 0;
for(int i = 1;i <= n;i ++)
        for(int j = a[i].d - 1;j >= a[i].t; j --)
                if (f[j] < f[j - a[i].t] + a[i].p){
                        f[j] = f[j - a[i].t] + a[i].p;
                        g[j].clear();
                        g[j] = g[j - a[i].t];
                        g[j].push_back(a[i].id);
                }
```

CF 864 E. Fire

```
int ans = 0;
for(int i = 1;i <= 2000;i ++)
        if (f[i] > f[ans]) ans = i;
printf( “%d\n” , f[ans]);
printf( “%d\n” , g[ans].size());
for(int i = 0;i < g[ans].size();i ++)
        printf( “%d  “, g[ans][i]);
```

POJ 1742 Coins

有 n 种面额的硬币。第 i 个硬币的面额是 v[i]，个数为 c[i].

问最多能搭配出多少种不超过 m 的金额。

n≤100，m≤100000，1≤v[i]≤100000，1≤c[i]≤1000

样例：(答案是8：1,2,3,4,5,6,7,8)

3 10

1 2 4

2 1 1

POJ 1742 Coins

可以当作多重背包来做。

但是因为这里只是问是否存在方案。

如果不行可以一个个试，不用枚举 c[i] 次。

这样时间复杂度就是 O(nm)

POJ 1742 Coins

```
for(int i = 1;i <= n;i ++) scanf(“%d”, &v[i]);
for(int i = 1;i <= n;i ++) scanf(“%d”, &c[i]);
memset(f, 0, sizeof(f));f[0] = 1;
for(int i = 1;i <= n;i ++){
        memset(sum, 0, sizeof(sum));
        for(int j = v[i];j <= m;j ++)
                if(!f[j] && f[j- v[i]] && sum[j- v[i]] <
c[i])
                        f[j] = 1, sum[j] = sum[j- v[i]] + 1;
}
```

下节课再见