**★★★★☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11400: Lighting System Design**

**解題者：黃皓群**

**解題日期：2025年5月01日**

**題意：**

給定 n 種燈泡，每種燈泡的電壓為 V，購買該電壓電源需花費 K 元 ，購買一盞該燈泡需花費 C 元，並且需要提供 L 盞。電壓具備向上兼容性：若選擇電壓 Vi 的電源與燈泡，便可同時滿足所有電壓 ≤ Vi 的需求 (需將這些低電壓燈泡整批換成 Vi 之規格)。所求為尋找最小成本之電源方案。

**題意範例**：

假設有三種不同規格的燈泡：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 編號 | 電壓 V  | 電源成本 K | 燈泡單價 C | 需求數量 L |
| 1 | 100 | 500 | 1 | 60 |
| 2 | 200 | 300 | 2 | 40 |
| 3 | 400 | 200 | 6 | 20 |

以下是部分可選擇的方案：

|  |  |  |
| --- | --- | --- |
| 方案 | 成本計算 | 電源成本 K |
| 200 V 供 100 + 200 V400 V 供 400 V | 300 + 2 \* (60 + 40) = 500200 + 6 \* 20 = 320 | **820** |
| 400 V 供 全部燈泡 | 200 + 6 \* (60 + 40 + 20) = 920 | 920 |
| 三種電源電壓各自供應 | 500 + 1 \* 60 = 560300 + 2 \* 40 = 380200 + 6 \* 20 = 320 | 1260 |

**解法：**

以 DP 解之

先將燈泡根據電壓排序，定義 DP 狀態：

dp [i] = 前1…i 種規格燈泡(電壓)之最小總成本

dp [0] = 0

狀態轉移式：

把第 j ~ i 個全部打包給電壓 i ，並承接前段最小成本

複雜度： O(n2), n ≤ 1000

**解法範例：**

見題意範例表格

**討論：**

(1) 窮舉法 O(2n-1), n ≤ 1000，複雜度過高。

(2) Greedy Method 當前最便宜的電壓可能阻斷之後用更高電壓「一次吃掉整段」的大幅省成本機會。

(3) 以前綴和方法加速計算區間燈泡數量

**程式：**

#include <bits/stdc++.h>

using namespace std;

struct bulb

{

    int v,k,c,l;

};

bool cmp(bulb a,bulb b)

{

    return a.v<b.v;

}

int main()

{

    int n;

    while(cin>>n && n)

    {

        bulb bulbs[n];

        for(int i=0;i<n;i++)

            cin>>bulbs[i].v>>bulbs[i].k>>bulbs[i].c>>bulbs[i].l;

        sort(bulbs,bulbs+n,cmp);

        int prefixL[n];

        prefixL[0]=bulbs[0].l;

        for(int i=1;i<n;i++)

            prefixL[i]=prefixL[i-1]+bulbs[i].l;

        int dp[n];

        memset(dp,INT\_MAX,sizeof(dp));

        dp[0]=bulbs[0].k+bulbs[0].c\*bulbs[0].l;

        for(int i=1;i<n;i++)

        {

            dp[i]=bulbs[i].k+bulbs[i].c\*prefixL[i];

            for(int j=0;j<i;j++)

                dp[i]=min(dp[i],dp[j]+bulbs[i].k+bulbs[i].c\*(prefixL[i]-prefixL[j]));

        }

        cout<<dp[n-1]<<endl;

    }

}