

P1478 陶陶摘苹果（升级版）

题目描述

又是一年秋季时，陶陶家的苹果树结了 nnn 个果子。陶陶又跑去摘苹果，这次他有一个 a 公分的椅子。当他手够不着时，他会站到椅子上再试试。

这次与 NOIp2005 普及组第一题不同的是：陶陶之前搬凳子，力气只剩下 s 了。当然，每次摘苹果时都要用一定的力气。陶陶想知道在 $s < 0$ 之前最多能摘到多少个苹果。

现在已知 n 个苹果到达地上的高度 x_i ，椅子的高度 a ，陶陶手伸直的最大长度 b ，陶陶所剩的力气 s ，陶陶摘一个苹果需要的力气 y_i ，求陶陶最多能摘到多少个苹果。

输入格式

第 1 行：两个数 苹果数 n ，力气 s 。

第 2 行：两个数 椅子的高度 a ，陶陶手伸直的最大长度 b 。

第 3 行~第 $3+n-1$ 行：每行两个数 苹果高度 x_i ，摘这个苹果需要的力气 y_i 。

输出格式

只有一个整数，表示陶陶最多能摘到的苹果数。

输入样例

```
8 15
20 130
120 3
150 2
110 7
180 1
50 8
200 0
140 3
120 2
```

输出样例

```
4
```

解析

尽可能的优先摘去消耗体力小的苹果，因此，我们首先需要对所有苹果的消耗力气进行排序。当然，要把高度不够的进行排除。

编码

```
#include <bits/stdc++.h>

using namespace std;
const int Maxn = 5000 + 1;
long long n, s, a, b, countNum = 0;

//定义苹果的结构体，高度和消耗体力
struct Apple {
    int high;
    int power;
};

Apple apples[Maxn];

//按照消耗的体力进行排序
bool Compare(Apple a, Apple b) {
    return a.power < b.power;
}

void Count() {
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        //能够得到，并且体力充足
        if (a + b >= apples[i].high && (sum + apples[i].power) <= s)
            //累计最后的结果
            countNum++;
        sum += apples[i].power;
    }
}

int main() {
    //读入样例数据
    scanf("%d %d", &n, &s);
    scanf("%d %d", &a, &b);
    for (int i = 0; i < n; ++i) {
        scanf("%d %d", &apples[i].high, &apples[i].power);
    }
    sort(apples, apples + n, Compare);
    Count();
    cout << countNum;
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。



