

P2705 小球

题目描述

有 R 个红色盒子和 B 个蓝色盒子，还有 R 个红色小球和 B 个蓝色小球。每个盒子只能装一个小球，每个小球都要放在一个盒子里。

如果把一个红色小球放在一个红色盒子里，那么得分是 C 。如果把一个蓝色小球放在一个蓝色盒子里，那么得分是 D 。如果把一个红色小球放在一个蓝色盒子里，那么得分是 E 。如果把一个蓝色小球放在一个红色盒子里，那么得分也是 E 。

现在给出 R, B, C, D, E 。应该如何放置这些小球进盒子，才能使得总得分最大？输出最大的总得分。

输入输出格式

输入格式：

一行，5 个整数，分别是 R, B, C, D, E 。

输出格式：

一个整数，最大总得分。

输入输出样例

输入样例#1：

复制

```
1 | 2 3 100 400 200
```

输出样例#1：

复制

```
1 | 1400
```

输入样例#2：

复制

```
1 | 2 3 100 400 300
```

输出样例#2：

复制

```
1 | 1600
```

说明

【数据规模】

$1 \leq R \leq 100$, $1 \leq B \leq 100$, $-1000 \leq C, D, E \leq 1000$ 。

思路

同学们用了数学的方法，确实，这题需要用数学的方法，

我写了记忆化搜索，发现记忆化的空间不够用，需要用 $100 * 100 * 100 * 100$ 会超内存，

用四维数组，用 10^6 开数组的话，只能开到32，因为 $(2^5)^4 = 2^{20}$ ，就是“兆”级别了

钱嘉欢 用的感觉像是贪心策略，思路也很清晰

周子涵 写的思路特别清晰，其他同学可以借鉴一下

昨天只有李明烨用了搜索的策略，要鼓励一下，

其他同学都用了数学的方法，都很棒！

下面是我的错误的，只能得到小数据的“记忆化搜索”的代码

代码（小部分分）

```
1 #include<iostream>
2 #include<cstring>
3 using namespace std;
4 long long r,b,c,d,e,h;
5 long long mm[32][32][32][32];
6 long long maxx(int a,int b,int c,int d){
7     int ans=a;
8     if(ans<b) ans=b;
9     if(ans<c) ans=c;
10    if(ans<d) ans=d;
11    return ans;
12 }
13 long long deal(int r_ball,int b_ball,int r_box,int b_box){
14     if(r_ball==0 && b_ball==0 && r_box==0 && b_box==0) return 0;
15     if(r_ball<0 || b_ball<0 || r_box<0 || b_box<0) return -99999999;
16     if(mm[r_ball][b_ball][r_box][b_box]!=-1) return mm[r_ball][b_ball][r_box][b_box];
17     long long ans1=c+deal(r_ball-1,b_ball,r_box-1,b_box);
18     long long ans2=d+deal(r_ball,b_ball-1,r_box,b_box-1);
19     long long ans3=e+deal(r_ball-1,b_ball,r_box,b_box-1);
20     long long ans4=e+deal(r_ball,b_ball-1,r_box-1,b_box);
21     long long ans=maxx(ans1,ans2,ans3,ans4);
22     mm[r_ball][b_ball][r_box][b_box]=ans;
23     return ans;
24 }
25 int main()
26 {
27     cin>>r>>b>>c>>d>>e; //输入
28     memset(mm,255,sizeof(mm));
29     cout<<deal(r,b,r,b); //输出
30     return 0;
31 }
32
```

一下是同学们正确的代码

```
1 //周子涵
2 #include<bits/stdc++.h>
3 using namespace std;
4 int r,b,c,d,e;
5 int main(){
6     cin>>r>>b>>c>>d>>e;
7     if(e*2>c+d){
8         if(r>b) cout<<(r-b)*c+b*2*e<<endl;
9         else cout<<(b-r)*d+r*2*e<<endl;
10    }else cout<<r*c+b*d<<endl;
11    return 0;
12 }
13 /*
14 从所有红球都在红框里，所有蓝球都在蓝框里
15 到尽量多的红球在蓝框里，尽量多的蓝球在红框里
16 就是先设置一个初始答案（第一种），
17 再把一个红球放进蓝框，把一个蓝球放进红框，以此类推
18 每次都在执行一个把答案-(c+d)+e*2的步骤
19 这个数是固定的，要么越加越大，要么越加越小
20 所以只要判断所有红球都在红框里，所有蓝球都在蓝框里
21 和尽量多的红球在红框里，尽量多的蓝球在蓝框里
22 这两种方法哪个好就用哪个
23 */
```

```
1 //叶炜辰
2 #include<bits/stdc++.h>
3 using namespace std;
4 int a,b,c,m,n;
5 int main(){
6     cin>>m>>n;
7     cin>>a>>b>>c;
8     if(a+b>=2*c){//判断颜色配对，分数更多
9         cout<<m*a+b*n;
10        return 0;
11    }
12    else{//判断颜色不配对，分数更多
13        if(m>=n){
14            cout<<2*n*c+a*(m-n);//输出
15            return 0;
16        }
17        else{
18            cout<<2*m*c+b*(n-m);//输出
19            return 0;
20        }//蓝盒子与红盒子的数量要分类讨论
21    }
22    return 0;
23 }
```

```
1 //钱嘉欢
2 #include<iostream>
3 using namespace std;
4 long long r,b,c,d,e,h;
5 //r是red简写，b是blue的简写，c、d、e是题目里的，h指和
6 int main()
7 {
```

```

8      cin>>r>>b>>c>>d>>e;//输入
9      while(c+d>=e*2&&r>0&&b>0)
10     {
11         r--;
12         b--;
13         h+=c+d;
14     }
15     //两个两个一组，如果两个分别放到同样的颜色里比放到不同的颜色里得分多，而且两个颜色的球数量都大于0，就
加得分
16     while(c+d<e*2&&r>0&&b>0)
17     {
18         r--;
19         b--;
20         h+=e*2;
21     }
22     //两个两个一组，如果两个放到不同的颜色里分比别放到同样的颜色里得分多，而且两个颜色的球数量都大于0，就
加得分
23     h+=c*r;
24     h+=d*b;
25     //把还没有放的小球放到同样的颜色格子里，然后加得分
26     //数学方法
27     cout<<h;//输出
28     return 0;
29 }
30

```

```

1  //李明烨
2  #include<cstdio>
3  #include<iostream>
4  #include<algorithm>
5  using namespace std;
6  long long red,blue,r_r,b_b,r_b,zg=-2147483647,r,b;
7  void check(long long sco){
8      zg=max(zg,sco);
9      return ;
10 }
11 void dfsb(long long now,long long s){
12     if(now==red+blue+1){
13         check(s);
14         return ;
15     }
16     if(r>0){
17         r--;
18         dfsb(now+1,s+r_b);
19         r++;
20     }
21     if(b>0){
22         b--;
23         dfsb(now+1,s+b_b);
24         b++;
25     }
26 }
27 void dfsr(long long now,long long s){
28     if(now==red+1){
29         dfsb(now,s);
30         return ;
31     }
32     if(r>0){

```

```

33     r--;
34     dfsr(now+1,s+r_r);
35     r++;
36 }
37 if(b>0){
38     b--;
39     dfsr(now+1,s+r_b);
40     b++;
41 }
42 }
43 int main(){
44     scanf("%1d%1d%1d%1d%1d",&red,&blue,&r_r,&b_b,&r_b);
45     r=red;b=blue;
46     dfsr(1,0);
47     printf("%1d\n",zg);
48     return 0;
49 }

```

```

1 //刘宇鑫
2 #include<bits/stdc++.h>
3 using namespace std;
4 int main(){
5     int r, b, c, d, e;
6     cin>>r>>b>>c>>d>>e;
7     int ans=0;
8     ans+=r*c+b*d;
9     //初始情况为红放红，蓝放蓝，然后交换
10    //交换后，红球收益由c变成e，蓝球收益由d变e
11    if(2*e > c+d){//如果换了得分高，就一直换
12        int Min = min(r, b);//但两者交换数量有限，只能取r和b中小的那个数
13        ans += Min * (2*e-c-d);
14    }
15    cout<<ans;
16    return 0;
17 }

```

```

1 //殷学楷
2 #include<bits/stdc++.h>
3 using namespace std;
4 long long r,b,c,d,e,ans,mn;
5 int main(){
6     cin>>r>>b>>c>>d>>e;
7     mn=min(r,b);//求出最小的
8     if(2*e>c+d){//玄学，如果两个e >c+d。因为两个e和c+d所用的小球，箱子个数相等；
9         if(r>=b) ans=mn*e*2+(r-b)*c;
10        if(b>r) ans=mn*e*2+(b-r)*d;
11    }
12    else ans=r*c+b*d;//否则用原来的方法，红放红，蓝放蓝；
13    cout<<ans;
14    return 0;
15 }
16

```