

## 题意

最近，阿Q开了一间宠物收养所。收养所提供两种服务：收养被主人遗弃的宠物和让新的主人领养这些宠物。

每个领养者都希望领养到自己满意的宠物，阿Q根据领养者的要求通过他自己发明的一个特殊的公式，得出该领养者希望领养的宠物的特点值 $a$ （ $a$ 是一个正整数， $a < 2^{31}$ ），而他也给每个处在收养所的宠物一个特点值。这样他就能够很方便的处理整个领养宠物的过程了，宠物收养所总是会有两种情况发生：被遗弃的宠物过多或者是想要收养宠物的人太多，而宠物太少。

1. 被遗弃的宠物过多时，假若到来一个领养者，这个领养者希望领养的宠物的特点值为 $a$ ，那么它将会领养一只目前未被领养的宠物中特点值最接近 $a$ 的一只宠物。（任何两只宠物的特点值都不可能是相同的，任何两个领养者的希望领养宠物的特点值也不可能是一样的）如果有两只满足要求的宠物，即存在两只宠物他们的特点值分别为 $a-b$ 和 $a+b$ 那么领养者将会领养特点值为 $a-b$ 的那只宠物。

2. 收养宠物的人过多，假若到来一只被收养的宠物，那么哪个领养者能够领养它呢？能够领养它的领养者，是那个希望被领养宠物的特点值最接近该宠物特点值的领养者，如果该宠物的特点值为 $a$ ，存在两个领养者他们希望领养宠物的特点值分别为 $a-b$ 和 $a+b$ ，那么特点值为 $a-b$ 的那个领养者将成功领养该宠物。

一个领养者领养了一个特点值为 $a$ 的宠物，而它本身希望领养的宠物的特点值为 $b$ ，那么这个领养者的不满意程度为 $\text{abs}(a-b)$

### 【任务描述】

你得到了一年当中，领养者和被收养宠物到来收养所的情况，希望你计算所有收养了宠物的领养者的不满意程度的总和。这一年初始时，收养所里面既没有宠物，也没有领养者。

## 思路

用set维护还未被选择的元素，因为店里同一时间人和狗不同时存在，所以用一个变量来标记当前店里的是人还是狗。

如果set为空或者当前新元素和店里的种类相同，就将该元素加入set，否则就查询最接近它的元素，用set的lower\_bound方法，删除选中的元素。

## 代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long LL;
4  typedef unsigned long long ULL;
5  typedef long double LDB;
6  set<int>s;
7  int now;
8  const int mod = 1000000;
9  LL read()
10 {
11     LL x=0,f=1;char ch=getchar();
12     while(ch<'0' || ch>'9'){if(ch=='-')f=-1;ch=getchar();}
13     while(ch>='0'&&ch<='9'){x=x*10+ch-'0';ch=getchar();}
14     return x*f;
15 }
16 int main()
17 {
18     int n;
19     n = read();
20     int a,b;
21     cin>>a>>b;
22     now = a;
23     s.insert(b);
24     int ans = 0;
25     set<int>::iterator it;
26     set<int>::iterator it2;
27     for(int i=2;i<=n;i++)
28     {
29         a = read();
30         b = read();
31         if(a == now)
32             s.insert(b);
33         else if(s.empty())
34         {
35             now = a;
36             s.insert(b);
37         }
38         else
39         {
40             it = s.lower_bound(b);
41             if(it == s.end())
42             {
43                 it -- ;
44                 ans+= abs(b - *it);
45                 s.erase(it);

```

```
46         }
47         else if(*it != b)
48         {
49             if(it == s.begin())
50             {
51                 ans+=abs(*it - b);
52                 s.erase(it);
53             }
54             else
55             {
56                 int p1 = *it;
57                 it2 = it;
58                 it2--;
59                 int p2 = *it2;
60                 if(p1 - b < b - p2)
61                 {
62                     ans+= (p1-b);
63                     s.erase(it);
64                 }
65                 else
66                 {
67                     ans+=(b-p2);
68                     s.erase(it2);
69                 }
70             }
71         }
72         else
73             s.erase(it);
74     }
75     ans %=mod;
76 }
77 printf("%d\n",ans);
78 }
```