首先要明白：此题最关键的是相邻两个的人交换对于前面的人答案没影响，而且对于后面的人答案也没有影响。也就是说相邻两人的位置交换只会对这两个人产生影响。那么我们就先考虑这两个人。

我们对于国王身后的两个点来分析

队列可能是这样的：

左 右

king: a0 b0

p1 a1 b1

​p2 a2 b2

​ 则ans1=max(a0/b1,a0\*a1/b2)

队列可能是这样的：

左 右

king: a0 b0

​p2 a2 b2

​p1 a1 b1

则ans2=max(a0/b2,a0\*a2/b1)

如果设ans1<ans2,因为a0\*a1/b2> a0/b2,所以可推得：a0\*a2/b1> a0\*a1/b2,得a1\*b1<a2\*b2

所以，为了ans取到最小值，我们需要将ai∗bi较小的放在前面。

因此将左手与右手的乘积从小到大排序，然后计算求最大值即可。注意要用高精度。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | //国王游戏  #include <bits/stdc++.h>  using namespace std;  struct data  {  int a,b;  } man[10005];  int CMP(data a,data b)  {  return a.a\*a.b<b.a\*b.b;  }  string MAX(string a,string b)  {  if(a.size()!=b.size())  return a.size()>b.size()?a:b;  return a>b?a:b;  }  string ToString(int num, string ans="")  {  for(; num; num/=10)  ans+=num%10+'0';  reverse(ans.begin(),ans.end());  return ans;  }  int temp[20005];  string Multi(string a,string b) //高精度乘  {  reverse(a.begin(),a.end());  reverse(b.begin(),b.end());  for(int i=0; i<a.size(); i++)  for(int j=0; j<b.size(); j++)  temp[i+j]+=(a[i]-'0')\*(b[j]-'0');  for(int i=0; i<a.size()+b.size(); i++)  if(temp[i]>=10)  {  temp[i+1]+=temp[i]/10;  temp[i]%=10;  }  string ans="";  int len=a.size()+b.size()-1;  for(; temp[len]==0; len--);  for(int i=len; i>=0; i--)  ans+=temp[i]+'0';  return ans;  }  int dividend[20005],result[20005];  string Div(string a,int b) //高精度除  {  for(int i=0; i<a.size(); i++)  dividend[i+1]=a[i]-'0';  for(int i=1,rem=0; i<=a.size(); i++)  {  result[i]=(rem\*10+dividend[i])/b;  rem=(rem\*10+dividend[i])%b;  }  int len=1;  string ans="";  for(; result[len]==0&& len<a.size(); ++len);  for(int i=len; i<=a.size(); i++)  ans+=result[i]+'0';  return ans;  }  int main()  {  int n;  scanf("%d",&n);  for(int i=0; i<=n; i++)  scanf("%d%d",&man[i].a,&man[i].b);  sort(man+1,man+n+1,CMP);  string ans="0";  string mul=ToString(man[0].a);  for(int i=1; i<=n; i++)  {  ans=MAX(ans,Div(mul,man[i].b));  mul=Multi(mul,ToString(man[i].a));  }  cout<<ans<<endl;  return 0;  } |