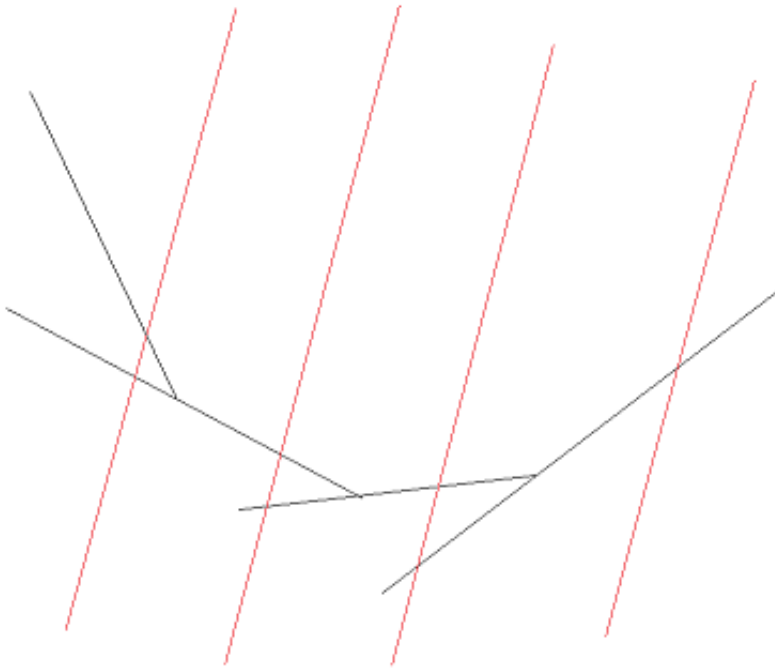


题意

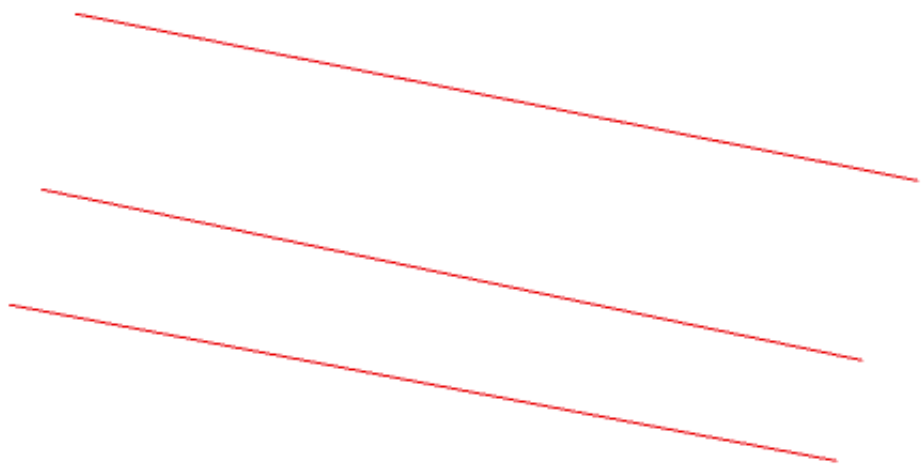
- 直线 $y = kx + b$
- 给出 n 条直线的 k b
- 所有直线不重合
- 求从 y 轴 $+\infty$ 方向往下看，能看到的直线有哪些

思路分析

最上方的轮廓线一定是内凹形状的，也就是所有能看到的线段，沿 x 轴，斜率逐渐增大
我们可以将可见线段用一个栈维护
沿斜率从小到大排序，一次插入



黑色线为目前已插入的可见线段，插入一条新的红线，如图，红线的位置如果与可见线段 x 相交，那么之前 x 以后的可见线段将全部不可见（出栈）



斜率相同的直线取最上面一条

注意边缘情况判断

代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  struct cre{
4      int k,b;
5      int id;
6  };
7  bool cmp(cre x,cre y)
8  {
9      if(x.k == y.k)
10         return x.b<y.b;
11
12         return x.k<y.k;
13  }
14  cre a[500005];
15  int st[500005];
16  int top = 0;
17  double x[500005];
18  int top2 = 0;
19  int ans[500005];
20  int main()
21  {
22      int n;
23      cin >> n;
24      for(int i = 1;i <= n;i++)
25      {
```

```

26     cin >> a[i].k >> a[i].b;
27     a[i].id = i;
28 }
29 sort(a+1,a+n+1,cmp);
30 if(n == 1)
31 {
32     cout<<"1"<<endl;
33     return 0;
34 }
35 if(n == 2)
36 {
37     cout<<"1 2"<<endl;
38     return 0;
39 }
40 int id1 = 1,id2 = 2;
41 while(a[id1].k == a[id2].k && id2<= n)
42 {
43     id1 = id2;
44     id2++;
45 }
46 if(id2>n)
47 {
48     cout<<a[n].id<<endl;
49     return 0;
50 }
51 st[++top] = id1;
52 st[++top] = id2;
53 x[++top2] = (a[id2].b - a[id1].b)*1.0 / (a[id1].k - a[id2].k);
54
55 for(int i = id2+1;i <= n;i++ )
56 {
57     int tt = st[top];
58     while(a[tt].k == a[i].k && top>0)
59     {
60         top2--;
61         top--;
62         tt = st[top];
63     }
64     double xx;
65     if(top>0)
66     {
67         xx = (a[i].b - a[tt].b)*1.0 / (a[tt].k - a[i].k) ;
68         while(xx <= x[top2] && top2>0)
69         {
70             top2--;
71             top--;

```

```
72         tt = st[top];
73         xx = (a[i].b - a[tt].b)*1.0 / (a[tt].k - a[i].k);
74     }
75 }
76 st[++top] = i;
77 if(top>=2)
78     x[++top2] = xx;
79 }
80 for(int i = 1;i<=top ; i++)
81 {
82     ans[i] = a[st[i]].id;
83 }
84 sort(ans+1,ans+top+1);
85 for(int i=1;i<=top;i++)
86 {
87     cout<<ans[i]<<" ";
88 }
89 }
```