

堆操作

```
#include<iostream>
#include<algorithm>
#include<queue>
using namespace std;
struct node{
    int x,y;
    bool operator<(const node &b) const
    {
        return x<b.x||x==b.x&& y<b.y; //大的在队首,若想相反改小于号
    }
}a[100];
priority_queue<node> pq;//优先队列，如此相当于大根二叉堆
int main()
{
    int n;cin>>n;
    int i;
    for(i=1;i<=n;i++)
    {
        cin>>a[i].x>>a[i].y;
        pq.push(a[i]);
    }
    while(pq.size())
    {
        cout<<pq.top().x<<" "<<pq.top().y<<endl;
        pq.pop();
    }
    return 0;
}
```

//给你一非降数列，以及一组查询，查询某一特定元素是否存在于数列之中，如果存在，则输出该元素首次出现的位置，否则输出"error"。

//it=s.lower_bound(5);cout<<*it<<endl;//大于等于 5 的值

//it=s.upper_bound(5);cout<<*it<<endl;//大于 5 的值

```
#include<cstdio>
#include<iostream>
#include<algorithm>
using namespace std;
int a[3000000];
int main()
{
    int n,m,i;
    while(scanf("%d%d",&n,&m)!=EOF)
```

```
{
    for(i=0;i<n;i++) scanf("%d",&a[i]);
    while(m--)
    {
        int x;
        scanf("%d",&x);
        int *p=lower_bound(a,a+n,x);
        if(p-a==n||*p!=x) printf("error\n");
        else printf("%d\n",p-a+1);
    }
}
return 0;
}
```