排序：

第一类排序：选择排序O（n2）、插入排序O（n2）、冒泡排序O（n2）

选择排序：

通过每次在未定队列中选出最大（或最小）的数放到指定位置上，循环多次来排序

插入排序：

通过每次在已定队列中选择当前需要插入数的位置，并且插入来排序

冒泡排序：

通过每次从未定队列的末尾（或者队首）经过比较来将当前较大数（或者较小数）向未定队列的队首（或者末尾）移动，循环多次来排序

第二类排序：堆排序O（nlogn）、归并排序O（nlogn）、快速排序O（nlogn）

堆排序：

通过每次将数插入堆，并且维护一个大顶堆（或者小顶堆），来排序

归并排序：

通过每次将未定数列进行对半划分，缩小需要比较的数据域，直到划分到一个小门槛时，在进行简单排序，最后通过整合两个已定队列来排序

快速排序：

通过每次选择一个标识数，将未定队列中比标识数小（或者大）的数放到标识数左边（或左边），比标识数大（或者小）的放到标识数的右边（或右边）。标识数左右两边在进行重复递归进行这一步骤

第三类排序：计数排序O（n+k）、基数排序O（n\*k）、桶排序O（n+k）

计数排序：

通过统计数值范围内未定队列中每个数的出现次数，在通过出现直接重组有序队列

基数排序：

从个十百位的低位开始向高位开始，每次将当前位上的数收集到0至9的队列中，在通过队列重组队列，循环以排序

桶排序：

通过搜索数据范围，生成一定数量的捅，将未定数列中的数按照大小关系放入桶中，在每个桶内排序完成后再通过桶重组数组

竞赛需要：

第K大数——快速排序

逆序对——归并排序：POJ - 2299

二分：

整数集合的二分：

查找>=x的最小数：

while(l < r)

{

int mid = (l + r) >> 1;

if(a[mid] >= x)

r = mid;

else

l = mid + 1;

}

return a[l];

查找<=x的最大值：

while(l < r)

{

int mid = (l + r) >> 1;

if(a[mid] <= x)

r = mid;

else

l = mid - 1;

}

return a[l];

实数域上的二分：

精度eps确定的二分：

一般需要保留K位小数时，精度eps取10^-(k+2)

while(l + 1e-5 < r)//三位小数

{

double mid = (l + r) / 2;//double类型不能<<或者>>

if(check(mid))

r = mid;

else

l = mid;

}

return l;

精度eps不确定的二分：

一般采用固定循环次数的方法来缩小目标范围

for(int i = 0;i < 100;i++)

{

double mid = (l + r) / 2;//double类型不能<<或者>>

if(check(mid))

r = mid;

else

l = mid;

}

return l;

三分确定单峰（单谷）曲线的极值：NC200211

while(l + 1e-5 < r)//单峰情况

{

double lmid = l + (r - l) / 3;

double rmid = r - (r - l) / 3;

if(check(lmid) > check(rmid))

r = rmid;

else

l = lmid;

}

return l;