二分查找：

|  |
| --- |
| /\*\*  \* 二分查找关键元素在有序数组中的首次出现位置。  \*/  int lowerBonder(int [] array , int k){  int low = 0;  int high = array.length-1;  while(low < high){  int mid = low +(high-low)/2;  if(array[mid] < k){  low = mid + 1;  }else{  high = mid;  }  }  if(low >= array.length || array[low]!=k){//也可以不加low >= array.length，因为low是逐1向上加，等于high时退出  return -1;  }else{  return low;  }  }  /\*\*  \* 二分查找关键元素在有序数组中的末次出现位置。  \*/  int upperbound(int [] array , int k){  int low = 0;  int high = array.length-1;  while(low < high){  int mid = low +(high-low+1)/2;  if(array[mid] > k){  high = mid -1;  }else{  low = mid;  }  }  if(low >= array.length || array[low]!=k){//也可以不加low >= array.length，因为low是逐1向上加，等于high时退出  return -1;  }else{  return low;  }  } |