第一个：递归算法

|  |
| --- |
| **public** **class** test {  **public** **int** first(**int** i) {  **int** sum=0;  **if**(i==1) **return** 1;  sum=sum+i\*first(i-1);  **return** sum;  }  **public** **static** **void** main(String[] args) {  **int** e=**new** test().first(4);  System.***out***.println(e);  }  } |

第二个：排序

（从小到大排序）（从大到小同理）

|  |
| --- |
| **public** **class** test1 {  **public** **int**[] two(**int**[] ob) {  **int** t;  **for**(**int** i=0;i<ob.length;i++) {  **boolean** dt=**false**;  **if**(!dt) {//加判断提高效率  **for**(**int** j=i+1;j<ob.length;j++) {  **if**(ob[i]>=ob[j]) {  t=ob[j];  ob[j]=ob[i];  ob[i]=t;  dt=**true**;  }  }}  }  **return** ob;  }  **public** **static** **void** main(String[] args) {  **int**[] b=**new** test1().two(**new** **int**[] {1,4,6,2,8,9,3,0});  System.***out***.println(Arrays.*toString*(b));  }  } |

第三个：查找

|  |
| --- |
| **public** **class** test2 {  /\*  \* 二分法查找  \*/  **public** **int** sle(**int**[] arr ,**int** value) {  **int** low=0;  **int** high=arr.length-1;  **while**(low<=high) {  **int** mid=(low+high)/2;  **if**(value==arr[mid]) {  **return** mid;  }  **if**(value>arr[mid]) {  low=mid+1;  }  **if**(value<arr[mid]) {  high=mid-1;  }    }  **return** -1;//没找到就返回-1    }    **public** **static** **void** main(String[] args) {  **int**[] tt=**new** **int**[] {1,2,3,4,5};  **int** t=(**int**)(**new** test2().sle(tt,4));  System.***out***.println(t);  }  } |