**算法训练 最小乘积(基本型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给两组数，各n个。  
　　请调整每组数的排列顺序，使得两组数据相同下标元素对应相乘，然后相加的和最小。要求程序输出这个最小值。  
　　例如两组数分别为:1 3　　-5和-2 4 1  
  
　　那么对应乘积取和的最小值应为：  
　　(-5) \* 4 + 3 \* (-2) + 1 \* 1 = -25

输入格式

　　第一个行一个数T表示数据组数。后面每组数据，先读入一个n，接下来两行每行n个数，每个数的绝对值小于等于1000。  
　　n<=8,T<=1000

输出格式

　　一个数表示答案。

样例输入

2

3

1 3 -5

-2 4 1

5

1 2 3 4 5

1 0 1 0 1

样例输出

-25

6

本题的C++参考代码如下：

#include<iostream>

#include<algorithm>

using namespace std;

int a[8],b[8];

int main()

{

int T,n;

int i,j;

cin>>T;

while(T--)

{

int sum=0;

cin>>n;

for(i=0;i<n;i++)

cin>>a[i];

for(i=0;i<n;i++)

cin>>b[i];

sort(a,a+n);

sort(b,b+n);

for(i=0;i<n;i++)

{

sum+=a[i]\*b[n-1-i];

}

cout<<sum<<endl;

}

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

void sort1(int \*a,int n)

{

int i,j;

int tmp;

for(i=0;i<n-1;i++)

for(j=0;j<n-1-i;j++)

if(a[j]>a[j+1])

{

tmp=a[j];

a[j]=a[j+1];

a[j+1]=tmp;

}

}

void sort2(int \*a,int n)

{

int i,j;

int tmp;

for(i=0;i<n-1;i++)

for(j=0;j<n-1-i;j++)

if(a[j]<a[j+1])

{

tmp=a[j];

a[j]=a[j+1];

a[j+1]=tmp;

}

}

int main(void)

{

int T;

int n,i;

int total;

int a[8],b[8],c[8];

scanf("%d",&T);

while(T)

{

total=0;

scanf("%d",&n);

for(i=0;i<n;i++)

scanf("%d",&a[i]);

for(i=0;i<n;i++)

scanf("%d",&b[i]);

sort1(a,n);

sort2(b,n);

for(i=0;i<n;i++)

c[i]=a[i]\*b[i];

for(i=0;i<n;i++)

total+=c[i];

printf("%d\n",total);

T--;

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void sortInt(int[] num){

int temp = 0;

for(int i=0;i<num.length-1; i++){

for(int j=i+1; j<num.length;j++){

if(num[i]<num[j]){

temp = num[i];

num[i] = num[j];

num[j] = temp;

}

}

}

}

public static void main(String[] args) throws Exception {

BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));

String temp = buf.readLine();

int t = Integer.parseInt(temp);

for(int k=0; k<t; k++){

temp = buf.readLine();

Integer n = Integer.parseInt(temp);

String[] tempNum = buf.readLine().split(" ");

int[] num1 = new int[n];

int[] num2 = new int[n];

for(int i=0; i<tempNum.length; i++){

num1[i] = Integer.parseInt(tempNum[i]);

}

tempNum = buf.readLine().split(" ");

for(int i=0; i<tempNum.length; i++){

num2[i] = Integer.parseInt(tempNum[i]);

}

sortInt(num1);

sortInt(num2);

int sum = 0;

for(int i=0; i<n; i++){

sum += num1[i]\*num2[n-1-i];

}

System.out.println(sum);

}

}

}