**算法训练 集合运算**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给出两个整数集合A、B，求出他们的交集、并集以及B在A中的余集。

输入格式

　　第一行为一个整数n，表示集合A中的元素个数。  
　　第二行有n个互不相同的用空格隔开的整数，表示集合A中的元素。  
　　第三行为一个整数m，表示集合B中的元素个数。  
　　第四行有m个互不相同的用空格隔开的整数，表示集合B中的元素。  
　　集合中的所有元素均为int范围内的整数，n、m<=1000。

输出格式

　　第一行按从小到大的顺序输出A、B交集中的所有元素。  
　　第二行按从小到大的顺序输出A、B并集中的所有元素。  
　　第三行按从小到大的顺序输出B在A中的余集中的所有元素。

样例输入

5  
1 2 3 4 5  
5  
2 4 6 8 10

样例输出

2 4  
1 2 3 4 5 6 8 10  
1 3 5

样例输入

4  
1 2 3 4  
3  
5 6 7

样例输出

1 2 3 4 5 6 7  
1 2 3 4

锦囊1

排序后处理。

锦囊2

先排序，对于每个集合的操作，都使用两个指针来指向排序后的集合，对于相同元素特别处理。

本题的C++参考代码如下：

#include<cstdio>

#include<algorithm>

using namespace std;

int ta[2005];

int main()

{

int n,m,a[2005],b[2005];

int i,j,k;

scanf("%d",&n);

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

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

scanf("%d",&m);

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

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

sort(a,a+n);

sort(b,b+m);

k=0;

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

{

for(j=0;j<m;j++)

{

if(a[i]<=b[j])

break;

}

if(j!=m)

{

if(a[i]==b[j])

{

ta[i]=-1;

k=1;

printf("%d ",a[i]);

}

}

}

if(k)

printf("\n");

i=0;j=0;

while(i<n&&j<m)

{

if(a[i]<b[j])

{

if(ta[i]!=-1)

printf("%d ",a[i]);

i++;

}

else

{

printf("%d ",b[j]);

j++;

}

}

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

{

if(ta[i]!=-1)

printf("%d ",a[i]);

}

for(j;j<m;j++)

{

printf("%d ",b[j]);

}

printf("\n");

k=0;

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

{

if(ta[i]!=-1)

{

k=1;

printf("%d ",a[i]);

}

}

if(k)

printf("\n");

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#define MAX\_N 1001

int n,m;

int Partition(int x[],int low,int high)

{

int pivotkey;

pivotkey = x[low];

while(low < high)

{

while(low<high && x[high]>=pivotkey)

high--;

x[low] = x[high];

while(low<high && x[low]<=pivotkey)

low++;

x[high] = x[low];

}

x[low] = pivotkey;

return low;

}

void QSort(int x[],int low,int high)

{

int pivotloc;

if(low < high)

{

pivotloc = Partition(x,low,high);

QSort(x,low,pivotloc-1);

QSort(x,pivotloc+1,high);

}

}

void jiao(int x[],int y[],int xx,int yy)

{

int i,j;

i = 1;

j = 1;

while(i<=xx && j<=yy)

{

if(x[i] == y[j])

{

printf("%d ",x[i]);

i++;

j++;

}

else if(x[i] < y[j])

i++;

else

j++;

}

printf("\n");

}

int bing(int x[],int y[],int z[],int xx,int yy)

{

int i,j,k;

i = 1;

j = 1;

k = 1;

while(i<=xx && j<=yy)

{

if(x[i] < y[j])

{

z[k] = x[i];

i++;

}

else if(x[i] > y[j])

{

z[k] = y[j];

j++;

}

else

{

z[k] = x[i];

i++;

j++;

}

k++;

}

if(i > xx)

while(j <= yy)

{

z[k] = y[j];

k++;

j++;

}

else

while(i <= xx)

{

z[k] = x[i];

k++;

i++;

}

return k;

}

void yu(int x[],int y[],int xx,int yy)

{

int i,j;

i = 1;

j = 1;

while(i<=xx && j<=yy)

{

if(x[i] == y[j])

{

i++;

j++;

}

else if(x[i] < y[j])

{

printf("%d ",x[i]);

i++;

}

else

j++;

}

if(j > yy)

while(i <= xx)

{

printf("%d ",x[i]);

i++;

}

printf("\n");

}

int main()

{

int i,l;

int a[MAX\_N],b[MAX\_N];

int c[2014];

scanf("%d",&n);

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

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

QSort(a,1,n);

scanf("%d",&m);

for(i=1; i<=m; i++)

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

QSort(b,1,m);

jiao(a,b,n,m);

l = bing(a,b,c,n,m);

for(i=1; i<l; i++)

printf("%d ",c[i]);

printf("\n");

yu(a,b,n,m);

return 0;

}

本题的Java参考代码如下：

import java.io.\*;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Iterator;

import java.util.Set;

public class Main {

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

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

int n = Integer.parseInt(br.readLine());

int[] arr = new int[n];

String st[] = br.readLine().split(" ");

for (int a = 0; a < arr.length; a++) {

arr[a] = Integer.parseInt(st[a]);

}

Arrays.sort(arr);

int m = Integer.parseInt(br.readLine());

int[] tag = new int[m];

String str[] = br.readLine().split(" ");

for (int a = 0; a < tag.length; a++) {

tag[a] = Integer.parseInt(str[a]);

}

Arrays.sort(tag);

func(arr, tag);

}

public static void func(int[] arr, int[] tag) {

int x;

for (int a = 0; a < arr.length; a++) {

x = Arrays.binarySearch(tag, arr[a]);

if (x >= 0) {

System.out.print(arr[a] + " ");

}

}

System.out.println();

Set<Integer> set = new HashSet<Integer>();

for (int a = 0; a < arr.length; a++) {

set.add(arr[a]);

}

for (int a = 0; a < tag.length; a++) {

set.add(tag[a]);

}

int[] sor = new int[set.size()];

Iterator<Integer> it = set.iterator();

while (it.hasNext()) {

for (int a = 0; a < sor.length; a++) {

sor[a] = it.next();

}

}

Arrays.sort(sor);

for (int a = 0; a < sor.length; a++) {

System.out.print(sor[a] + " ");

}

System.out.println();

int y;

for (int a = 0; a < arr.length; a++) {

y = Arrays.binarySearch(tag, arr[a]);

if (y < 0) {

System.out.print(arr[a] + " ");

}

}

System.out.println();

}

}