代码：

#include <iostream>

using namespace std;

#define ALEN 5

#define BLEN 10

int mmm(int\*& arr, int left, int right) {

int mid = arr[left];

while (left < right) {

while (left < right && arr[right] >= mid) {

right--;

}

arr[left] = arr[right];

while (left < right && arr[left] <= mid) {

left++;

}

arr[right] = arr[left];

}

arr[left] = mid;

return left;

}

void Sort(int\*& arr, int left, int right) {

if (left < right) {

int index = mmm(arr, left, right);

Sort(arr, left, index - 1);

Sort(arr, index + 1, right);

}

}

int inline rand(int n)

{

return rand() % n;

}

void Initnumber(int \* &a, int n) {

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

a[i] = rand(10\*n);

}

Sort(a, 0, n-1);

}

void Combine(int\*& arr, int\*& brr, int\*& crr) {

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

crr[i] = arr[i];

}

for (int j = ALEN; j <(ALEN+BLEN); j++) {

crr[j] = brr[j-ALEN];

}

Sort(crr, 0, (ALEN+BLEN) - 1);

}

int DeleteRepetition(int\*& crr,int Len) {

int n = 0;

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

if (crr[i] == crr[i + 1]) {

for (int j = i; j < Len -n- 1; j++) {

crr[j] = crr[j + 1];

}

n++;

i--;

}

}

return n;

}

int main()

{

srand(time(0));

int \* arr=new int[ALEN];

int \* brr=new int[BLEN];

int \* crr = new int[(ALEN+BLEN)];

Initnumber(arr, ALEN);

Initnumber(brr, BLEN);

Combine(arr, brr, crr);

int n = DeleteRepetition(crr, ALEN + BLEN);

cout << "arr is " << endl;

for (int i = 0; i < ALEN; i++)cout << arr[i]<<" ";

cout << endl<< "brr is " << endl;

for (int j = 0; j < BLEN; j++)cout << brr[j]<<" ";

cout << endl<<"combination is " << endl;

for (int k = 0; k < (ALEN+BLEN-n); k++)cout << crr[k] << " ";

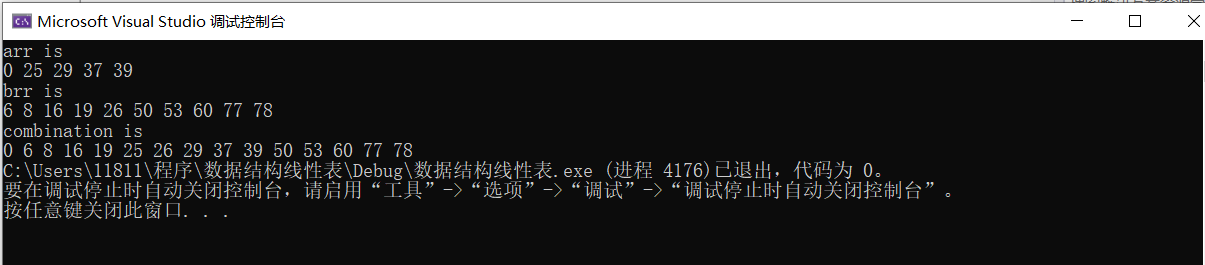
delete[]arr;

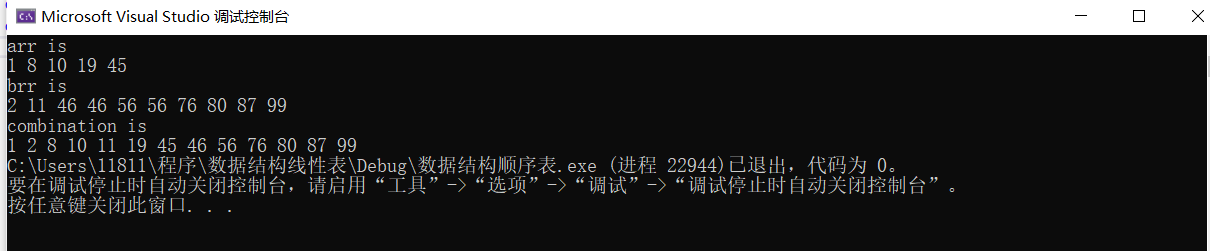
delete[]brr;

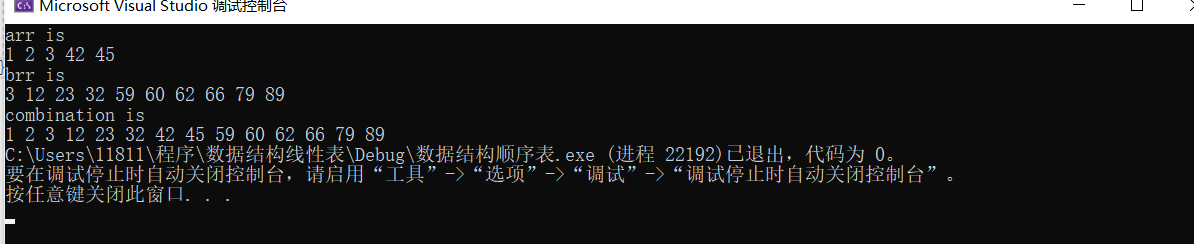
delete[]crr;

}

运行结果：







时间复杂度：

排序算法采用快速排序，平均时间复杂度为O(n\*logn)

为了简化，以下过程用A代替ALEN，B代替BLEN

初始化数组时间复杂度分别为A+A\*log(A)和B+B\*log(B)

混合数组时间复杂度为A+ B+(A+B)\*log(A+B)

删除一个重复的平均时间复杂度（第i位和后一位重复）：

+2=

假设出现重复的个数等可能，出现重复的位置等可能，

则平均重复数为，删除重复平均复杂度

输出时间复杂度为A+ B+ A+B-

总时间复杂度为+A\*log(A)+B\*log(B) +(A+B)\*log(A+B)+

即为O(A+B)+O(A\*logA)+O(B\*logB)+O((A+B)\*log(A+B))+O((A+B)2)

所以时间复杂度为O()