**DAA画–画Lab画3画–画January画25**

**Vishal画Gauba**

**1410110501**

***Source画Code:***

#include "stdio.h"

#include "stdlib.h"

#include "math.h"

#include "time.h"

#include "assert.h"

int画result[100000];

void画bubble\_sort(int\*画array,画int画n){

printf("Bubbling画the画array...\n");

int画c,d,swap;

for画(c画=画0画;画c画<画(画n画-画1画);画c++){

for画(d画=画0画;画d画<画n画-画c画-画1;画d++){

画画画画画画if画(array[d]画>画array[d+1]){

swap画画画画画画画=画array[d];

array[d]画画画=画array[d+1];

array[d+1]画=画swap;

}

}

画画 }

}

int画largest(int画a[],画int画n){

画画画画int画large画=画a[0],画i;

画画画画for(i画=画1;画i画<画n;画i++)

画画画画{

画画画画画画画画if(large画<画a[i])

画画画画画画画画画画画画large画=画a[i];

画画画画}

画画画画return画large;

}

void画radix\_sort(int画\*a,画int画n)画{

int画i,画b[100000],画m画=画0,画exp画=画1;

for画(i画=画0;画i画<画n;画i++)画{

if画(a[i]画>画m)

画画画m画=画a[i];

}

while画(m画/画exp画>画0)画{

int画box[10]画=画{0};

for画(i画=画0;画i画<画n;画i++)

画画画box[a[i]画/画exp画%画10]++;

for画(i画=画1;画i画<画10;画i++)

画画画box[i]画+=画box[i画-画1];

for画(i画=画n画-画1;画i画>=画0;画i--)

画画画b[--box[a[i]画/画exp画%画10]]画=画a[i];

for画(i画=画0;画i画<画n;画i++)

画画画a[i]画=画b[i];

exp画\*=画10;

#ifdef画SHOWPASS

画画printf("\n\nPASS画画画:画");

print(a,画n);

#endif

}

}

void画merge(画int画arr[],画画int画start1,画画int画end1,画画int画start2,画画int画end2){

画画画int画i,j,k;

画画画

画画画k画=画start1;

画画画i画=画start1;

画画画j画=画start2;

画画画//画while画there画are画elements画in画both画lists,画keep画picking画the画smaller画one

画画画while画((i<=end1)画&&画(j<=end2))

画画画{

画画画画画if画(arr[i]画<=画arr[j])

画画画画画{

画画画画画画画画result[k]画=画arr[i];

画画画画画画画画i++;

画画画画画画画画k++;

画画画画画}

画画画画画else

画画画画画{

画画画画画画画画result[k]画=画arr[j];

画画画画画画画画j++;

画画画画画画画画k++;

画画画画画}

画画画}

画画画//画one画of画the画lists画would画have画been画finished.画append画the画other画list画at画end

画画画//画Remaining画part画of画List画1

画画画while画(i画<=画end1)

画画画{

画画画画画画result[k]画=画arr[i];

画画画画画画i++;

画画画画画画k++;

画画画}

画画画//画Remaining画part画of画List画2

画画画while画(j画<=画end2)

画画画{

画画画画画画result[k]画=画arr[j];

画画画画画画j++;

画画画画画画k++;

画画画}

画画画assert(i==end1+1);

画画画assert(j==end2+1);

画画画assert(k==end2+1);

画画画//画Copy画the画merged画list画back画to画the画input画array

画画画for(i=start1;画i画<=画end2;画i++)

画画画画画arr[i]画=画result[i];

}

void画recursive\_mergesort(画int画arr[画],画画int画start,画画int画end){

画画画int画mid;

画画画int画n;画//画number画of画items

画

画画n画=画end-start+1;画

画画mid画=画(start画+画end)画/画2;

画画画if画(n画<画2)

画画画画画画return;

画画画//画recursive\_sort画each画part.画merge画the画two画sorted画parts

画画画recursive\_mergesort(arr,画start,画mid);

画画画recursive\_mergesort(arr,画mid+1,画end);

画画画//画merge画the画two画sorted画parts

画画画merge(arr,画start,画mid,画mid+1,画end);

}

Int main (int argc, char const \*argv[])

{

clock\_t t0 = clock();

int n=10000, i, \*array = (int\*) malloc(n\*sizeof(int));

printf("Generating the array...\n");

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

array[i] = rand()%n;

}

// bubble\_sort(array, n);

radix\_sort(array, n); /\* change 2D bucket array size in the function too \*/

// recursive\_mergesort(array, 0, n-1); /\* change global variable result too \*/

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

// printf("%d\n",array[i]);

// }

clock\_t t1 = clock();

printf("Time taken to execute = %lf\n",(double)(t1 - t0) / CLOCKS\_PER\_SEC);

return 0;

}

***Analysis画Table:***

**Time画is画in画seconds.**

|  |  |  |  |
| --- | --- | --- | --- |
| Input画Size | Bubble画Sort画time | Merge画Sort画time | Radix画Sort画time |
| 103 | 00.015625 | 0 | 0 |
| 104 | 00.265625 | 00.015625 | 00.015625 |
| 105 | 30.531250 | 00.234375 | 00.125000 |