B04902017 李立譽 資工一

2.1.(1):

cin>>n;

float matrix[n][n];

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

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

cin>>matrix[i][j];

}

}

float temp;

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

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

temp = matrix[i][j];

matrix[i][j] = matrix[j][i];

matrix[j][i] = temp;

}

}

2.1.(2):

if there is a lower triangular matrix A[n][n],we only store the element which index i >= j,and skip the upper triangular area(0).

getting:

matrix[n][n] -> array[(n^2 + n)/2]; (convert matrix into one dimension array)

k = 0;

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

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

if(i >= j){

array[k] = matrix[i][j];

k++;

}

}

}

putting:

to print matrix[i][j] from array[(n^2 + n)/2];

if(i < j) printf("0")l

else {

printf("%d\n",array[ (i\*(i + 1))/2 + j]);

}

2.2.(1)

I use two vectors to store the data. One is sorted by user->item->time->result, the other is sorted by item->user->time->result.

The first vector(sorted by user) is convenient to operate function "accept","item", and the second vector is convenient to operaate

function "user","ratio","findtime\_item".

Because function "ratio" need the number of being recommended,and no matter the parameter is, it use the same data. Thus, I store

the number of being recommended in the vector of each users to reduce working time.

2.2.(2)

53 secs,linux working station

2.2.(3)

I use the testdata on the Facebook to test my program.

First, I input the testdata and output the result to a ".out" file.

Second, I use diff to make sure my answer is correct.

Third, I check the user time and real time to make sure my program is efficient.