TEST REPORT

* **TEST CASES**

1. INPUT

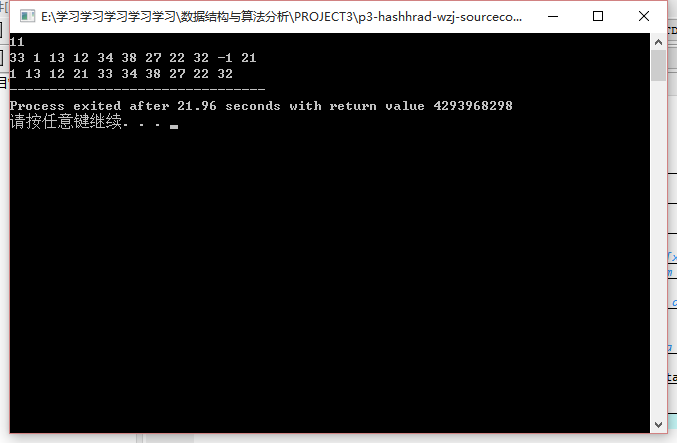
11

33 1 13 12 34 38 27 22 32 -1 21

OUTPUT

1 13 12 21 33 34 38 27 22 32

RUNNING RESULT



1. INPUT

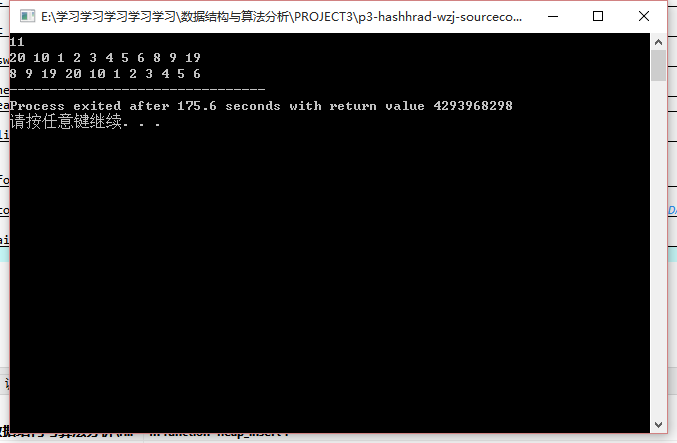
11

20 10 1 2 3 4 5 6 8 9 19

OUTPUT

8 9 19 20 10 1 2 3 4 5 6

RUNNING RESULT



1. INPUT

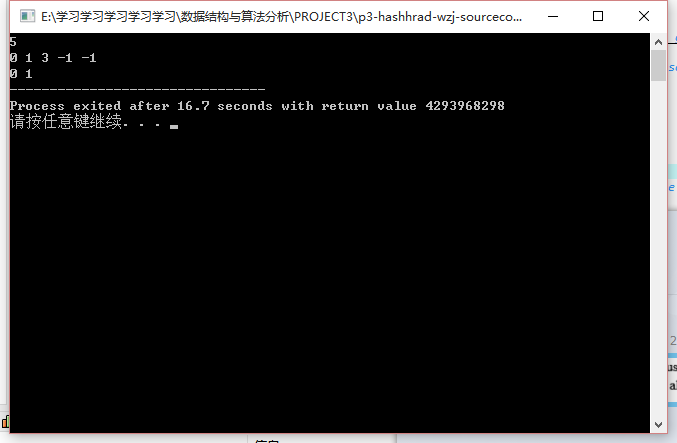
5

0 1 3 -1 -1

OUTPUT

0 1

RUNNING RESULT



* **ANALYSIS AND COMMENTS**

TIME COMPLEXITY: The time complexity is mainly depend on the function *formmap* .

void formmap(ptr\* map,int\* tab,int n,struct heaptype\* heap){//draw a DAG according the given sequence

int i,j;

int hash;

int nodetome;

ptr newcell;

for(i=0;i<n;i++) {//initiate all nodes of the map **O(N)**

newcell=(ptr)malloc(1\*sizeof(ptr));

newcell->n\_edges\_in=0;newcell->n\_edges\_out=0;

newcell->nextnodes=NULL;

map[i]=newcell;

}

for(i=0;i<n;i++){ **O(N)**

if(tab[i]<0){

map[i]->n\_edges\_in=-998998;//don't worry, the in-degree -998998 won't change later

continue;

}

hash=tab[i]%n;

if(hash==i) {//if there is no collision

heap\_insert(heap,i,tab);

continue;

}

//if there exist collision

map[i]->n\_edges\_in=(i-hash+n)%n;//change the node i's in-degree in one step

for(j=0;j<map[i]->n\_edges\_in;j++){ **O(N)**

nodetome=(hash+j)%n;

linknodeab\_onlydealwitha(map,nodetome,i);

}

}

}

As we can see, the time complexity of this program is O(N2).

COMMENT:

The wonder of this problem is the topological sorting. The solution of this problem can inspire us that Graph Theory is always important in computer science.