Smart elephant

201632739 한국 알고리즘 과제

Code:

#include <stdio.h>

#include <stdlib.h>

void sortW(int arr[][2] ,int size,int\*\*\*re)

{

int i,max,j,w,s,in;

int\*\* ne = (int\*\*)malloc(sizeof(int\*)\*size);

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

{

ne[i] = (int\*)malloc(sizeof(int)\*3);

}

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

{

ne[i][2] = i;

ne[i][1] = arr[i][1];

ne[i][0] = arr[i][0];

}

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

{max=0;

for(j=1;j<size-i;j++)

{

if(ne[max][0]<=ne[j][0])

{

if(ne[max][0] == ne[j][0])

{

if(ne[max][1]>ne[j][1])

{

}

else

{

max=j;

}

}

else

{

max = j;

}

}

}

w = ne[size-i-1][0];

s = ne[size-i-1][1];

in = ne[size-i-1][2];

ne[size-i-1][1] = ne[max][1];

ne[size-i-1][0] = ne[max][0];

ne[size-i-1][2] = ne[max][2];

ne[max][1] = s;

ne[max][0] = w;

ne[max][2] = in;

}

\*re = ne;

}

void sortS(int arr[][2] ,int size, int\*\*\*re)

{

int i,max,j,w,s,in;

int\*\* ne = (int\*\*)malloc(sizeof(int\*)\*size);

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

{

ne[i] = (int\*)malloc(sizeof(int)\*3);

}

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

{

ne[i][2] = i;

ne[i][1] = arr[i][1];

ne[i][0] = arr[i][0];

}

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

{max=0;

for(j=1;j<size-i;j++)

{

if(ne[max][1]>= ne[j][1])

{

if(ne[max][1] == ne[j][1])

{

if(ne[max][0]<ne[j][0])

{

}

else

{

max=j;

}

}

else

{

max = j;

}

}

}

w = ne[size-i-1][0];

s = ne[size-i-1][1];

in = ne[size-i-1][2];

ne[size-i-1][1] = ne[max][1];

ne[size-i-1][0] = ne[max][0];

ne[size-i-1][2] = ne[max][2];

ne[max][1] = s;

ne[max][0] = w;

ne[max][2] = in;

}

\*re = ne;

}

int judge(int\*a,int\*b)

{

if(a[0] == b[0])

{

if(a[1] == b[1])

{

return 1;

}

}

return 0;

}

int max(int a,int b)

{

if(a>b)

return a;

return b;

}

void lts(int arr[][2],int size)

{

int\*\* w; int\*\* s;

int table[size+1][size+1];

int i,j;

int maxl;

sortW(arr,size,&w);

sortS(arr,size,&s);

for(i=0;i<size+1;i++)

{

for(j=0;j<size+1;j++)

{

table[i][j] = 0;

}

}

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

{

for(j=1;j<size+1;j++)

{

if(judge(w[i-1],s[j-1]))

{

table[i][j] = table[i-1][j-1]+1;

}

else

{

table[i][j] = max(table[i-1][j],table[i][j-1]);

}

}

}

printf("%d\n",table[size][size]);

maxl = table[size][size];

int a = size; int b=size;

int list[maxl]; int count=0;

while((a!=0)&&(b!=0))

{

if(table[a][b] == table[a-1][b])

{

a = a-1;

}

else

{

if(table[a][b]==table[a][b-1])

{

b=b-1;

}

else

{

list[count] = s[b-1][2];

a=a-1; b=b-1;

count+=1;

}

}

}

for(i=maxl-1;i>=0;i--)

{

printf("%d\n",list[i]+1);

}

}

int main(void)

{

int arr[9][2]={{6008,1300},{6000,2100},{500,2000},{1000,4000},{1100,3000},{6000,2000},{8000,1400},{6000,1200},{2000,1900}};

int arr1[6][2] = {{6000,2100},{1000,4000},{8000,1400},{2000,1900},{6008,1300},{500,2000}};

int arr2[8][2] = {{6008,1300},{6000,2100},{500,2000},{1000,3500},{1100,4000},{8000,1400},{3000,1500},{2000,1900}};

lts(arr,9);

printf("\n\n");

lts(arr1,6);

printf("\n\n");

lts(arr2,8);

return 0;

}

Result:

스크린샷, 컴퓨터, 모니터, 실내이(가) 표시된 사진

자동 생성된 설명