201735861 이승화

소스 코드

인풋파일 : input.txt

#include <stdio.h>

#include <stdlib.h>

#define DEBUG

typedef struct

{

int size;

int IQpoint;

int IQSt;

int num;

} elephant;

elephant\* input[1001];

int\*\* arr;

int\*\* arr\_up;

int table\_row\_size = 0;

int input\_size = 1;

void AddInput(int esize, int IQ)

{

elephant\* temp = (elephant\*)malloc(sizeof(elephant));

temp->size = esize;

temp->IQpoint = IQ;

temp->num = input\_size;

input[input\_size] = temp;

input\_size++;

}

int\*\* Newtable(int arrsize, int row\_count)

{

int\*\* table = (int\*\*)malloc(sizeof(int\*) \* row\_count);

for (int i = 0; i < row\_count; i++)

{

table[i] = (int\*)malloc(sizeof(int) \* arrsize);

for (int j = 0; j < arrsize; j++)

{

table[i][j] = 0;

}

}

return table;

}

void file\_open()

{

FILE\* file = fopen("input.txt", "r");

elephant temp;

while (fscanf(file, "%d %d", &(temp.size), &(temp.IQpoint)) == 2)

{

AddInput(temp.size, temp.IQpoint);

};

fclose(file);

}

void sort()

{

// 이 함수는 몸무게 내림차순, (같은 몸무게일때는 IQ의 역순으로 정렬한다)

for (int i = 1; i < input\_size; i++)

{

for (int j = 1; j < input\_size - i; j++)

{

if (input[j]->IQpoint < input[j + 1]->IQpoint)

{

elephant\* temp = input[j];

input[j] = input[j + 1];

input[j + 1] = temp;

}

}

}

int a = 0;

for (int i = 1; i < input\_size; i++)

{

if (i != 1 && input[i - 1]->IQpoint == input[i]->IQpoint)

input[i]->IQSt = a;

else

input[i]->IQSt = ++a;

}

a += 1;

table\_row\_size = a;

for (int i = 1; i < input\_size; i++) input[i]->IQSt = a - input[i]->IQSt;

for (int i = 1; i < input\_size; i++)

{

for (int j = 1; j < input\_size - i; j++)

{

if (input[j]->size < input[j + 1]->size)

{

elephant\* temp = input[j];

input[j] = input[j + 1];

input[j + 1] = temp;

}

}

}

}

void print\_debug()

{

printf("\r\nTable\r\n");

printf(" null |");

printf(" null |");

for (int i = 1; i < input\_size; i++)

{

printf(" %4d, %4d |", input[i]->size, input[i]->IQSt);

}

printf("\r\n");

for (int i = 0; i < table\_row\_size; i++)

{

printf(" %4d |", i);

for (int j = 0; j < input\_size; j++)

{

const char\* temp = 0;

switch (arr\_up[i][j])

{

case 5:

temp = "↖";

break;

case 4:

temp = "←";

break;

case 8:

temp = "↑";

break;

default:

temp = " ";

break;

}

printf(" %s %d |", temp, arr[i][j]);

}

printf("\r\n");

}

}

void print\_result()

{

// 순서를 반대로 따라간다.

int x = input\_size - 1;

int y = table\_row\_size - 1;

#ifdef DEBUG

printf("Max SIZE\r\n%4d\r\n\r\nSequence\r\n", arr[y][x]);

#else

printf("%d\r\n", arr[y][x]);

#endif // DEBUG

while (arr\_up[y][x] != 0)

{

switch (arr\_up[y][x])

{

case 5:

#ifdef DEBUG

printf("%4d (SIZE:%4d, IQ:%4d)\r\n", input[x]->num, input[x]->size, input[x]->IQpoint);

#else

printf("%d\r\n", input[x]->num);

#endif // DEBUG

y--; x--;

break;

case 4:

x--;

break;

case 8:

y--;

break;

}

}

}

void main()

{

file\_open();

sort();

arr = Newtable(input\_size, table\_row\_size);

arr\_up = Newtable(input\_size, table\_row\_size);

for (int i = 1; i < table\_row\_size; i++)

{

for (int j = 1; j < input\_size; j++)

{

if (i == input[j]->IQSt)

{

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

arr\_up[i][j] = 5;

}

else

{

if (arr[i - 1][j] < arr[i][j - 1])

{

arr[i][j] = arr[i][j - 1];

arr\_up[i][j] = 4;

}

else

{

arr[i][j] = arr[i - 1][j];

arr\_up[i][j] = 8;

}

}

}

}

print\_result();

#ifdef DEBUG

print\_debug();

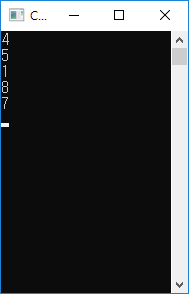
#endif // DEBUG

}

Input1

|  |  |
| --- | --- |
| 2600 | 1200 |
| 4500 | 2300 |
| 8900 | 6500 |
| 5100 | 9800 |
| 2300 | 2500 |
| 2600 | 500 |
| 9800 | 900 |
| 5400 | 1000 |

Output1



Input2

500 9100

3200 800

3200 500

200 300

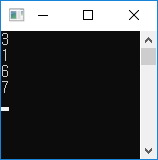
300 200

900 500

1500 200

9800 1000

Output2



Input3

500 300

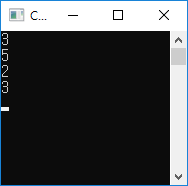
200 400

300 200

600 800

100 500

Output3



소스 상단에

#define DEBUG

를 정의할경우  
  
