Data Structure

Homework 4

Deadline: 2021/01/09 Sat. 11:55

Task 1: (I/O: 15 points, coding style: 5 points)

Given a number *n* (*n* = -1 or 1 ≤ *n* ≤ 2^{10}) represents *n* non-repeating integers. The next row is given *n* positive integers (1 ≤ values ≤ 2^{31} - 1). Please implement quick sort to sort the given numbers in the ascending order. Your program has to read till `-1`. The program must be implemented **by quick sort**, or you will get zero points.

Example:

Input	Output

`5`
`5 1 3 4 2`
	`> 1 2 3 4 5`

`10`
`6 8 7 10 1 5 2 4 9 3`
	`> 1 2 3 4 5 6 7 8 9 10`

`-1`	

Task 2: (I/O: 25 points, coding style: 5 points)

Given an undirected simple graph with n (n = -1 or $1 \le n \le 2^{10}$) vertices and m (m = -1 or $1 \le m \le 2^{10}$) edges. The vertices are labeled from 0 to (n - 1). Each pair of data refers to an edge which connects two vertices. Please write a program to show that:

1. Whether the vertices can be drawn in two colors, with no two adjacent vertices have the same color.
2. Use the adjacency matrix to represent the given graph.

Your program has to read till `-1`. Note that you need to use the **adjacency matrix** to implement the **depth first search**, or you will get zero points.

Example:

Input	Output

3 3
0 1
1 2
2 0
	> Not bicolorable
	> 0 1 1
	> 1 0 1
	> 1 1 0

9 8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
	> Bicolorable
	> 0 1 1 1 1 1 1 1 1
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0
	> 1 0 0 0 0 0 0 0 0

-1 -1	

Task 3: (I/O: 25 points, coding style: 5 points)

Given a 01-matrix with n ($n = -1$ or $1 \le n \le 2^{10}$) rows and m ($m = -1$ or $1 \le m \le 2^{10}$) columns to represent a water pipe map. The first row of the 01-matrix will have only one “1” to refer to the starting point. Please write a program to show that the time of water goes. Your program has to read till `-1`. Note that you need to use the **adjacency list** to implement the **breadth first search**, or you will get zero points.

Example:

Input	Output

4 6
0 0 0 1 0 0
0 0 1 1 1 0
1 1 1 0 1 0
0 1 0 0 0 0
	> 0 0 0 1 0 0
	> 0 0 3 2 3 0
	> 6 5 4 0 4 0
	> 0 6 0 0 0 0

5 7
1 0 0 0 0 0 0
1 1 1 1 1 1 0
0 1 0 0 1 0 1
1 1 1 0 1 1 1
1 0 1 0 1 1 1
	> 1 0 0 0 0 0 0
	> 2 3 4 5 6 7 0
	> 0 4 0 0 7 0 11
	> 6 5 6 0 8 9 10
	> 7 0 7 0 9 10 11

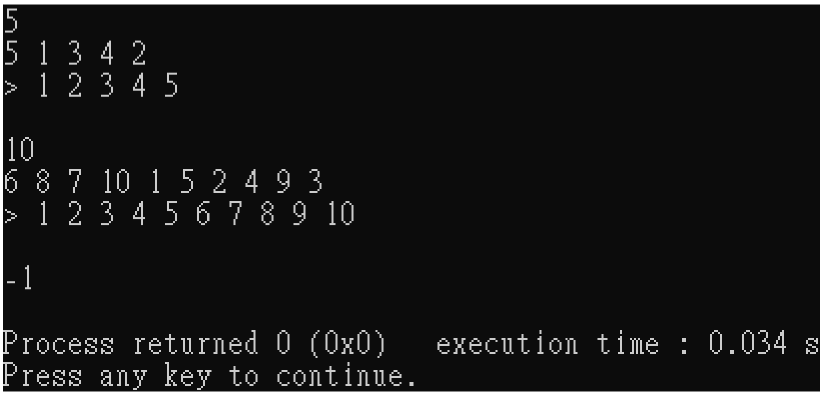
6 6
0 1 0 0 0 0
1 1 0 0 1 1
0 1 1 1 1 0
1 1 1 1 1 1
0 0 0 0 1 0
0 0 0 1 1 0
	> 0 1 0 0 0 0
	> 3 2 0 0 **7** **8**
	> 0 3 4 5 6 0
	> 5 4 5 6 7 8
	> 0 0 0 0 8 0
	> 0 0 0 10 9 0

-1 -1	

Put the files below in the folder (folder name: studentID), and compress this folder as **“studentID.zip”**.
1. Two source code files (filename: studentID_1.c, studentID_2.c)
2. One report with your coding environment (OS, IDE, ...), problems you encountered, and references. (filename: studentID.pdf) (10 points)
All the file names are correct, or you’ll get zero points. (10 points)
You must hand in the assignment on time, or you will get zero points.
Warning: We encourage you to discuss assignments with each other. However, you have the responsibility to finish the assignments individually. **Do not copy others’ assignment, or you will get zero points.**

Expected result:

(1)



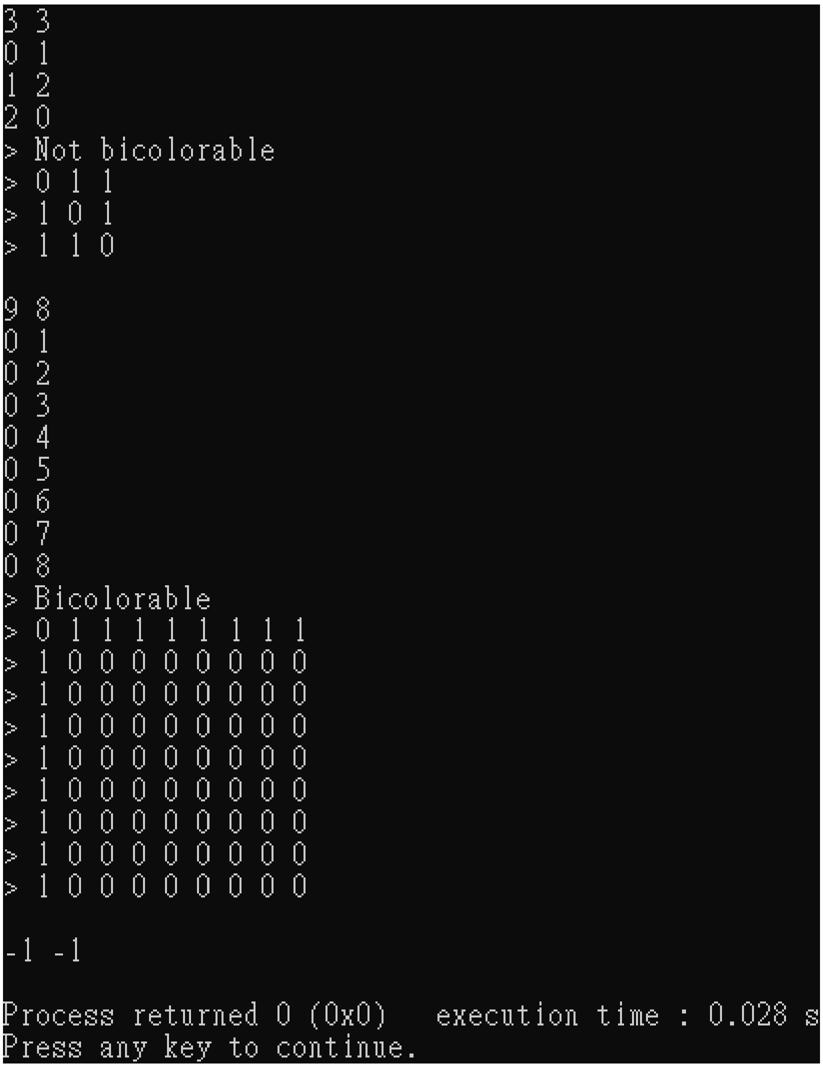
```
5
5 1 3 4 2
> 1 2 3 4 5

10
6 8 7 10 1 5 2 4 9 3
> 1 2 3 4 5 6 7 8 9 10

-1

Process returned 0 (0x0)   execution time : 0.034 s
Press any key to continue.
```

(2)



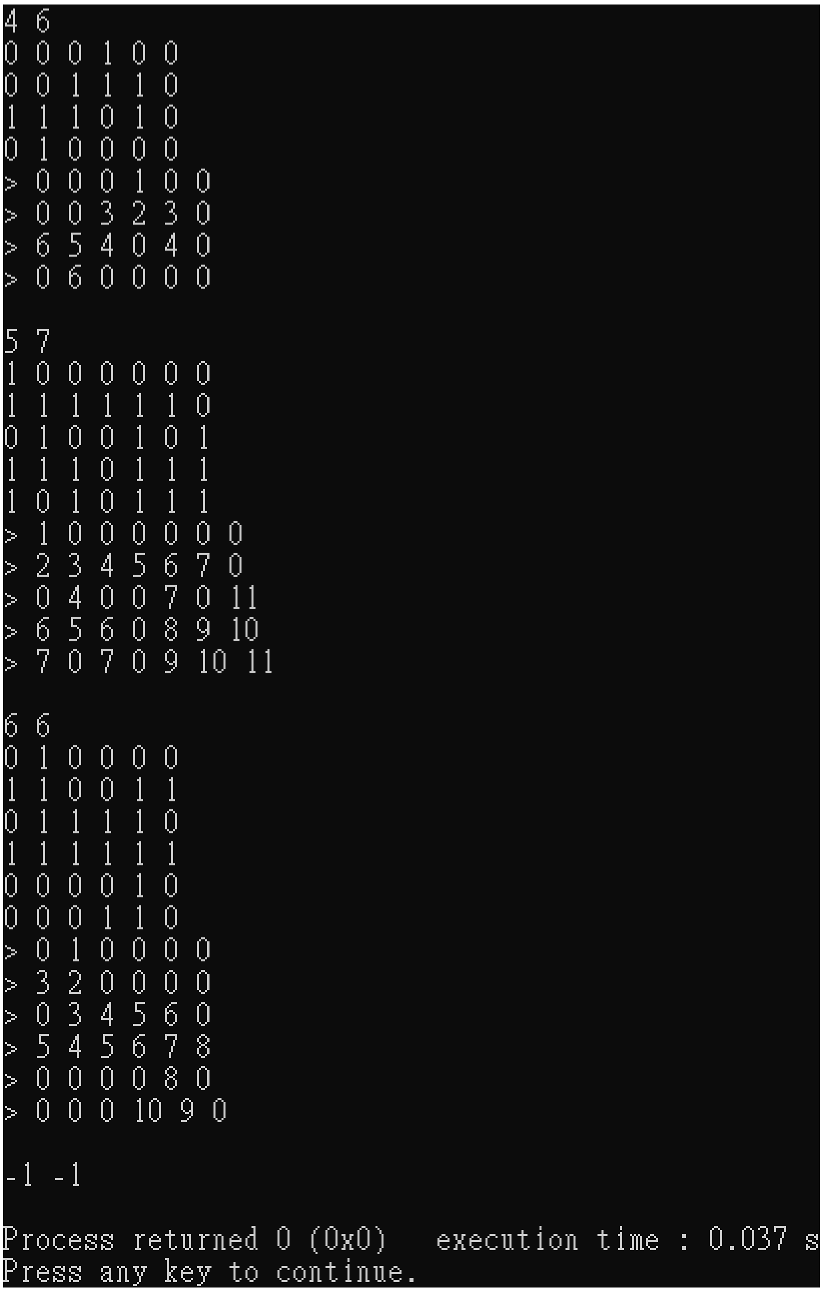
```
3 3
0 1
1 2
2 0
> Not bicolorable
> 0 1 1
> 1 0 1
> 1 1 0

9 8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
> Bicolorable
> 0 1 1 1 1 1 1 1 1
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0
> 1 0 0 0 0 0 0 0 0

-1 -1

Process returned 0 (0x0)   execution time : 0.028 s
Press any key to continue.
```

(3)



```
4 6
0 0 0 1 0 0
0 0 1 1 1 0
1 1 1 0 1 0
0 1 0 0 0 0
> 0 0 0 1 0 0
> 0 0 3 2 3 0
> 6 5 4 0 4 0
> 0 6 0 0 0 0

5 7
1 0 0 0 0 0 0
1 1 1 1 1 1 0
0 1 0 0 1 0 1
1 1 1 0 1 1 1
1 0 1 0 1 1 1
> 1 0 0 0 0 0 0
> 2 3 4 5 6 7 0
> 0 4 0 0 7 0 11
> 6 5 6 0 8 9 10
> 7 0 7 0 9 10 11

6 6
0 1 0 0 0 0
1 1 0 0 1 1
0 1 1 1 1 0
1 1 1 1 1 1
0 0 0 0 1 0
0 0 0 1 1 0
> 0 1 0 0 0 0
> 3 2 0 0 0 0
> 0 3 4 5 6 0
> 5 4 5 6 7 8
> 0 0 0 0 8 0
> 0 0 0 10 9 0

-1 -1

Process returned 0 (0x0)   execution time : 0.037 s
Press any key to continue.
```