

Data Structure

# Lab 3. Queue

Shin Hong

Apr 17, 2020



# Submission & Evaluation

- Submission deadline: 4 PM, Apr 21 (Tue)
- Online test: <http://hisolve.handong.edu> Session ID: 2xDE5Yr
- Evaluation: 5 points x 1 problem (total 5 points)
  - 5 points/problem: Succeed on time (before the deadline)
  - 4 points/problem: Succeed by 9 PM, Apr 24 (Thur)
  - Upto 3 points/problem: submit a report by 4 PM, Apr 25 (Fri)
- Note
  - You must use `queue.c` used in the class  
<https://github.com/hongshin/DataStructure/blob/queue/ver2/queue.c>
  - You must construct a solution as a single source code file

# Island (1/2)



- Write a program that reads a map of island(s) and then prints out the number of islands, the minimum and the maximum areas of the islands
- Preliminaries
  - A map is a 2D array where each cell indicates whether the corresponding coordinate is land (1) or ocean (0).
  - A man can move from a cell to another cell when both cells are of land and they share one border (i.e., adjacent)
  - An island is a maximum set of land cells where a man can reach from one member cell to another member cell
  - The area of an island is the number of member cells

# Input and Output

4

- Input
  - Given from the standard input
  - First line contains two integers  $W$  and  $H$  for  $1 \leq W \leq 20$  and  $1 \leq H \leq 20$ .  $W$  represents the width and  $H$  the height of the map.
  - From second line to  $(H+1)$ -th line, each line has  $W$  integers to represent the map. The  $x$ -th number at  $(y+1)$ -th line represents whether the  $(y, x)$  cell is land or ocean
  - The map contains at least one island
- Output
  - Print the number of islands, the minimum area among the islands and the maximum area among the islands
  - Make sure to put newline (`'\n'`) at the end

```
8 8 ↵
0 1 1 0 0 1 1 0 ↵
0 1 1 0 0 0 0 0 ↵
0 0 1 0 0 1 1 0 ↵
0 1 0 0 0 1 1 0 ↵
0 1 1 0 0 1 1 0 ↵
0 1 1 0 1 1 1 0 ↵
0 1 1 0 0 1 1 0 ↵
0 1 1 0 0 1 1 0 ↵
```

<Input I>

```
4 2 13 ↵
```

<Output I>

Lab 3.  
Queue

Data Structure

2020-04-17

# Example

5

0	1	1	0	0	1	1	0
0	1	1	0	0	0	0	0
0	0	1	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	1	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0