SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Stack and Queue questions

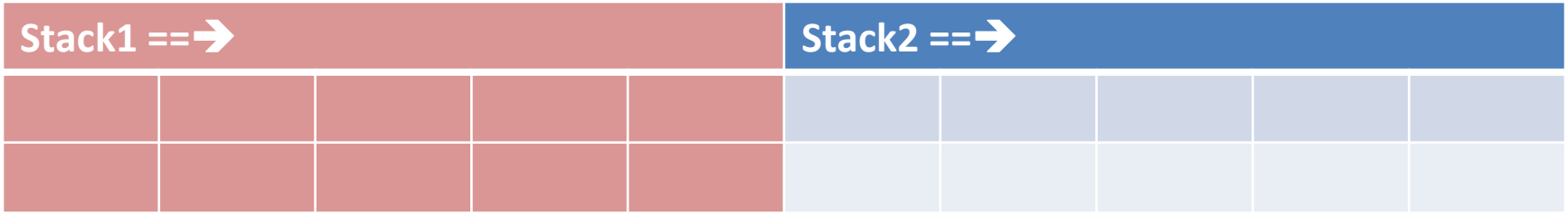
swatimali@somaiya.edu

Somaiya TRUST

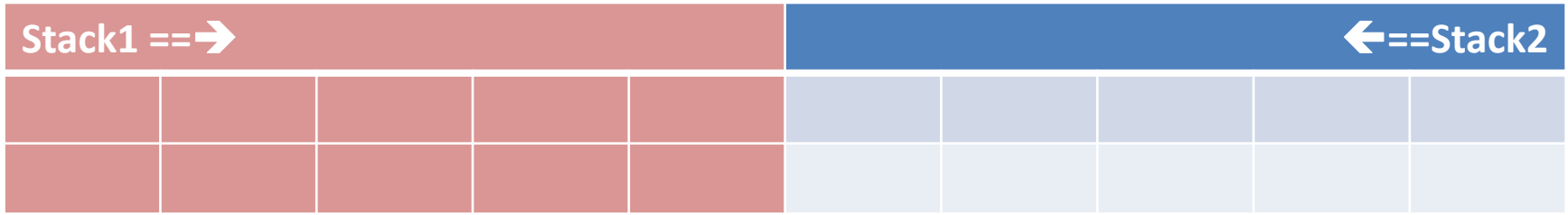
Question 1

- Implement two stacks in one array

Two stacks in one array



OR



Question 2

- Implement queue using stacks .
- How many minimum number of queues will be needed for this operation?

Question 2-solution1

- **Method -1 (Making enQueue operation costly)**
- This method makes sure that oldest entered element is always at the top of stack 1, so that deQueue operation just pops from stack1. To put the element at top of stack1, stack2 is used.

- ***enQueue(q, x):***
- *While stack1 is not empty, push everything from stack1 to stack2.*
- *Push x to stack1 (assuming size of stacks is unlimited).*
- *Push everything back to stack1.*

- ***deQueue(q):***
- *If stack1 is empty then error*
- *Pop an item from stack1 and return it*

Question 2-solution2

- **Method 2: (Making deQueue operation very costly)**
- In this method, in en-queue operation, the new element is entered at the top of stack1. In de-queue operation, if stack2 is empty then all the elements are moved to stack2 and finally top of stack2 `is returned.
- ***enQueue(q, x)***
  - *Push x to stack1 (assuming size of stacks is unlimited).*
- ***deQueue(q)***
  - *If both stacks are empty then error.*
  - *If stack2 is empty, While stack1 is not empty, push everything from stack1 to stack2.*
  - *Pop the element from stack2 and return it.*

Solution 1 Vs solution2

- Method 2 is definitely better than method 1.
- Method 1 moves all the elements twice in enQueue operation, while method 2 (in deQueue operation) moves the elements once and moves elements only if stack2 empty.

Question 3

- Implement stack using queue.
- How many minimum number of stacks will be needed for this operation?

Question 4

- Problem 2- Undo-Redo feature in text editor/browser etc.
- Since the general expectation is that:
  - we undo actions in the *reverse* order in which they were done, and
  - we *redo* actions that we undo in the reverse order as well

Solution – using stacks

- Maintain two stacks- undo stack and redo stack
- every new action you do is pushed on the undo stack,
- every undo pops an item off the undo stack and pushes it onto the redo stack,
- every redo pops an item off the redo stack and pushes it onto the undo stack.

Question

There are a number of students in a school who wait to be served. Two types of events, ENTER and SERVED, can take place which are described below.

– ENTER: A student with some priority enters the queue to be served.

– SERVED: The student with the highest priority is served (removed) from the queue.

Each student has a name, CGPA. An unique id is assigned to each student entering the queue. The queue serves the students based on the following criteria (priority criteria):

1. The student having the highest Cumulative Grade Point Average (CGPA) is served first.
2. Any students having the same CGPA will be served in ascending order of the id.

- Input - The first line contains an integer, , describing the total number of events. Each of the subsequent lines will be of the following two forms:
  - ENTER name CGPA id: The student to be inserted into the priority queue.
  - SERVED: The highest priority student in the queue was served.
- **Output -** Prints the names of the students yet to be served in the priority order. If there are no such student, then the code prints EMPTY

Question

- **Sample Input**
- 12
- ENTER John 3.75 50
- ENTER Mark 3.8 24
- ENTER Shafaet 3.7 35
- SERVED
- SERVED
- ENTER Samiha 3.85 36
- SERVED
- ENTER Ashley 3.9 42
- ENTER Maria 3.6 46
- ENTER Anik 3.95 49
- ENTER Dan 3.95 50
- SERVED
- **Sample Output 0**
- Dan
- Ashley
- Shafaet
- Maria

Queries?

Thank you!