5118006-03 Data Structures

Queue

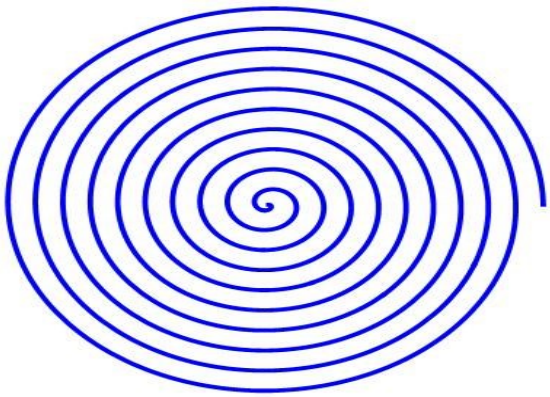
29 Mar 2024

Shin Hong

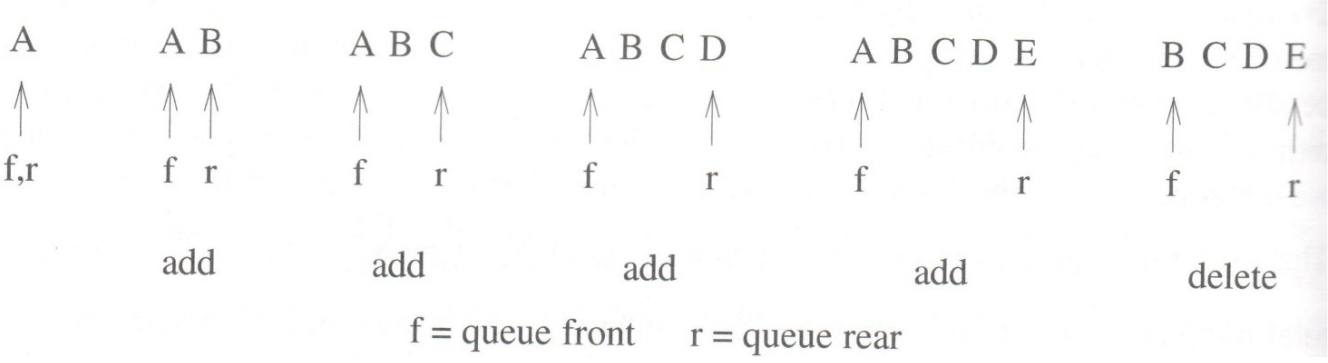
Queue

- A queue is an ordered list where insertion and deletion take place at different ends
  - insertion at the rear, deletion at the front

- A queue is known as a First-In-First-Out (FIFO) list



Example


A
A B
A B C
A B C D
A B C D E
B C D E
f,r
f r
f r
f r
f r
f r
add
add
add
add
delete
f = queue front
r = queue rear

Queue ADT

- Structure
  - **elements**: an array to hold elements
  - **capacity**: the maximum number of elements that the queue can hold
    - a queue can be bounded or unbounded

- Operations
  - **add(e)**: insert a new element **e** to if the queue is not full (i.e., enqueue)
  - **delete()**: return the least recently inserted element if the queue is not empty (i.e., dequeue)
  - **isEmpty()** : return whether the queue has at least one element or not
  - **isFull()** : return whether the queue is full or not

Circular Queue

- Implement a queue as an array-based list
  - maintain the indices of the front and the rear
  - the indices rotates around the indices of the element array

- Two designs for identifying empty and full states
  1. keep at least one element between front and rear
     - front indicates the next element to be removed if it is the same as rear (i.e., there exists at least one element)
     - rear indicates the empty slot for the next coming element
  2. store the number of elements currently in a queue (or whether the queue is full/empty or not)
     - front indicates the next element to be removed if there exists at least one element
     - rear indicates the empty slot for the next coming element if the queue is not full

Example. The Siege of Yodfat

- A puzzle composed by Claude Gaspar Bachet, inspired by the story of the siege of Yodfat

- Initially, n soldiers form a circle, and in turn, from left to right, every m-th soldier will be killed and removed from the circle until only one remains.
  - name the n soldiers as 1, 2, ... n.
  - The turn begins at 1, and moves to the next existing number, and when the turn reaches to the greatest, it comes backs to the smallest.

- Write a program that receives n and m, and finds the last remaining solider.

Example. Counting Islands (1/2)

- Write a program that reads a map of island(s) and then prints out the number of islands
  - a map is a 2D array where each cell represents whether the corresponding coordinate is land (1) or ocean (0).
  - two cells belong to the same island if and only if they share one border (i.e., adjacent)

- Input
  - first line contains two integers, W and H that represent the width and height of map, respectively, for $1 \leq W \leq 20$ and $1 \leq H \leq 20$.
  - 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

Example. Counting Islands (2/2)

```
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 ↲
```