

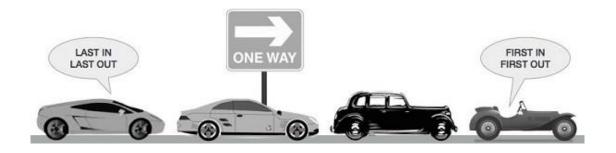
BITS F232: FOUNDATIONS OF DATA STRUCTURES & ALGORITHMS (1ST SEMESTER 2023-24) QUEUE ADT

Chittaranjan Hota, PhD Sr. Professor of Computer Sc. BITS-Pilani Hyderabad Campus hota[AT]hyderabad.bits-pilani.ac.in

SOLVING A MAZE USING A STACK

Enter a rectangular maze using the following characters: m - entry e - exit 1 - wall 0 - passage Enter one line at at time; end with Ctrl-z: 1111111 1111111 1110011 1100011 1111111	1111111 1111111 1110011 11.0011 1m.1e01 1111111 1111111 1111111 111111 111111	1111111 1110011 1101011 1m.1e01 1111111 1111111 1110011 1m.1e01 1111111
---	---	---

QUEUES





Img. Source: https://www.javascripttutorial.net/

Some Applications of Queues:

- 1. Network routers
- 2. Scheduling a single shared resource (CPU)
- 3. Concurrent web servers

Is it NOT a linear data structure?

What operations would you like to see in a queue ADT?

EXAMPLE SERIES OF OPERATIONS ON A QUEUE

ARRAY-BASED IMPLEMENTATION OF A QUEUE

Approach 1: Similar to stack based implementation where Q[0] be the front of the queue and have the queue grow from there.

How good is this approach?

Approach 2: Using an Array with three variables to avoid moving objects once they are placed in the queue.

Use three variables: f, r, and n. Let us see the dequeue and enqueue operations... (What is the complexity?)

If we repeatedly enqueue and dequeue a single element, what problem it might cause?

CONTINUED...

Approach 3: Use a <u>circular array</u> with 'f' and 'r' indices wrapping around the end of the queue.

OPERATIONS USING CIRCULAR ARRAY & INTERFACE

```
Algorithm size()
return n

Algorithm empty()
return (n == 0)
```

```
Algorithm dequeue ()

if empty() then

throw QueueEmpty

else

???

???
```

```
Algorithm enqueue (P) {
  if size() == N - 1 then
    throw QueueFull
  else {
    ???
    ???
    ???
```

```
template <typename E>
class Queue {
public:
  int size() const;
  bool empty() const;
  const E& front() const
    throw(QueueEmpty);
  void enqueue (const E& e);
  void dequeue()
    throw(QueueEmpty);
```