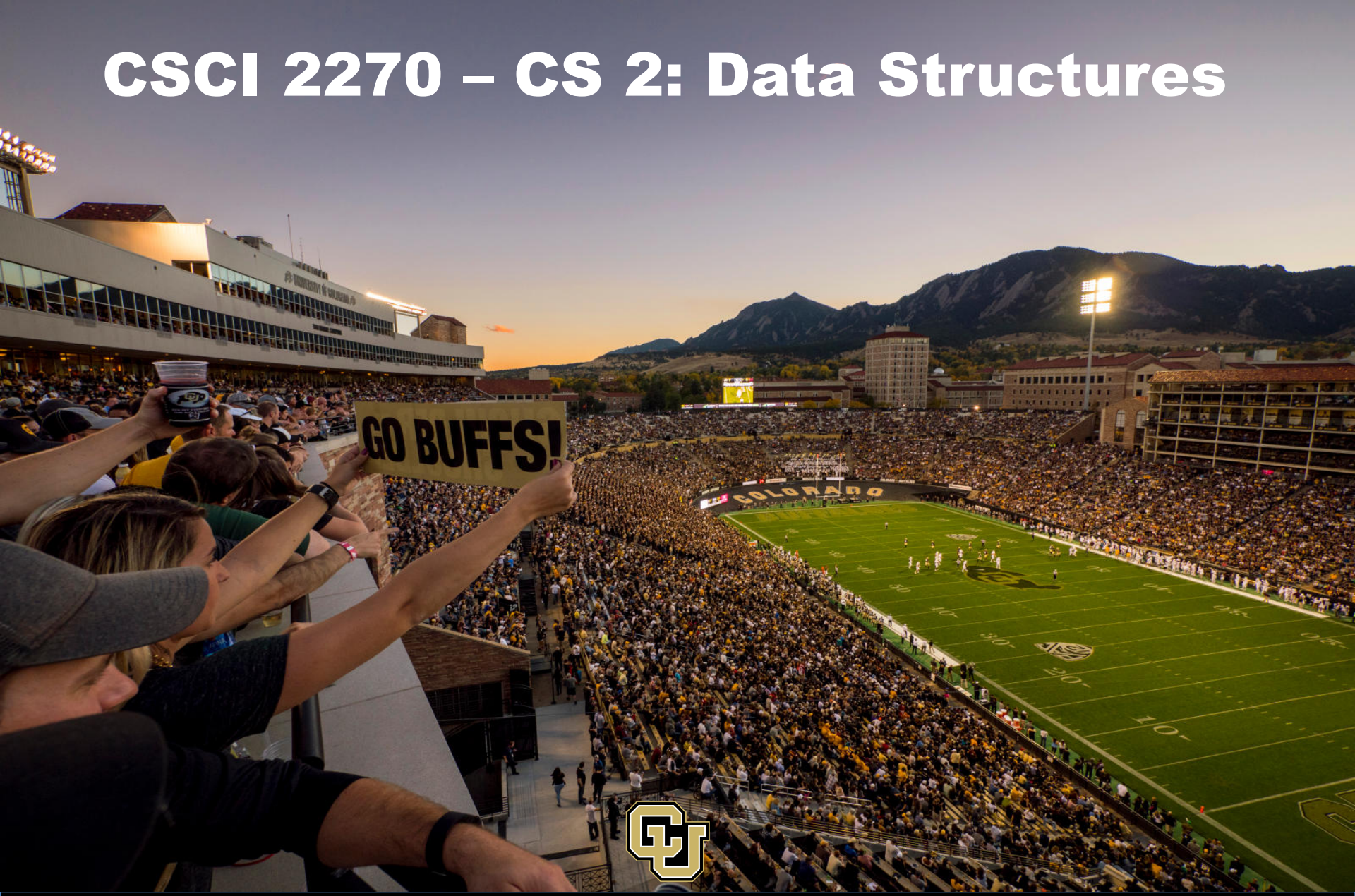
CSCI 2270 – CS 2: Data Structures



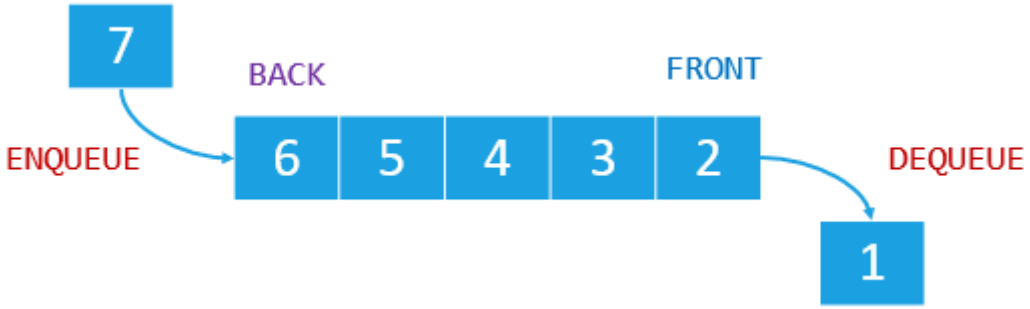
University of Colorado
Boulder

Topics

- Queues

Queues

- Similar to queue at a bank or grocery store
- Commonly used in computer simulation
- First-In First-Out (FIFO) data structure
- Elements are added at one end (back/rear) and deleted from the other end (front).
- Rear is accessed when adding. Front is used for deleting.



Queue Applications

- Commonly used in computer simulation
- CPU and disk scheduling
- Print spooling
  - When a number of print jobs is placed in a queue
- Call center phone systems

Queue ADT

private:

head – first item in the queue (next thing to be processed)

tail – last item in the queue

queueSize – # of elements currently in the queue

public:

initialize() – constructor – initialize queue to empty state

bool = isFull() – check if stack is full

bool = isEmpty() – check if queue is empty

enqueue(item) – add element to rear of queue

item = dequeue() – removes front element from the queue

peek() or front() – get element at front without deleting it

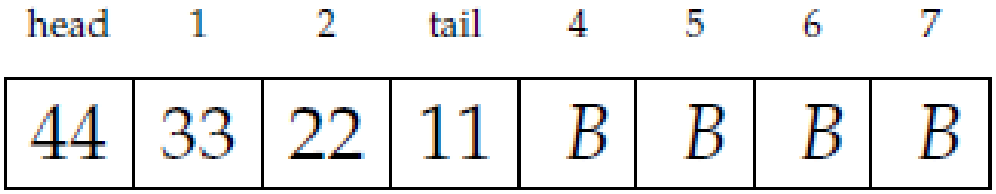
Queue Rules

- A queue may be implemented to have a bounded capacity (i.e. array).
  - If a queue is full and doesn't contain enough space for enqueue, the result will be queue overflow.
  - If you attempt to dequeue from an empty queue, queue underflow will occur.

Queue Implementations

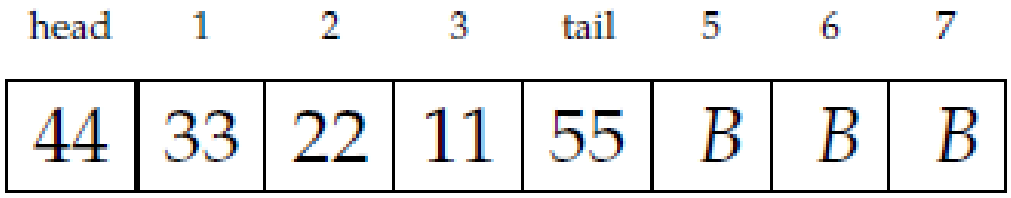
- Queues using linked lists
- Queue using simple arrays
- Queue using circular arrays
  - There isn't an end to the queue and elements are added or deleted in a circular motion.
  - The last position (last node) is connected back to the first position (first node) to make a circle.

Queues – Circular Arrays



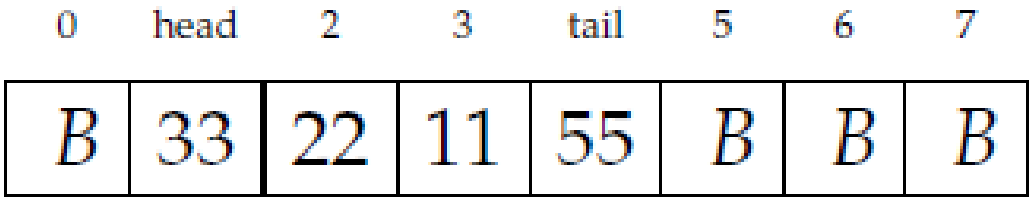
– enqueue(55);
– dequeue();
– enqueue(66); enqueue(77); enqueue(88);
– dequeue();
– enqueue(99);

Queues – Circular Arrays



– **enqueue(55);**
– dequeue();
– enqueue(66); enqueue(77); enqueue(88);
– dequeue();
– enqueue(99);

Queues – Circular Arrays



– enqueue(55);

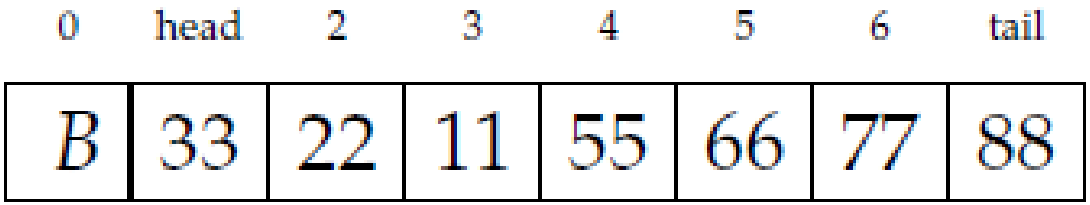
– **dequeue();**

– enqueue(66); enqueue(77); enqueue(88);

– dequeue();

– enqueue(99);

Queues – Circular Arrays



– enqueue(55);

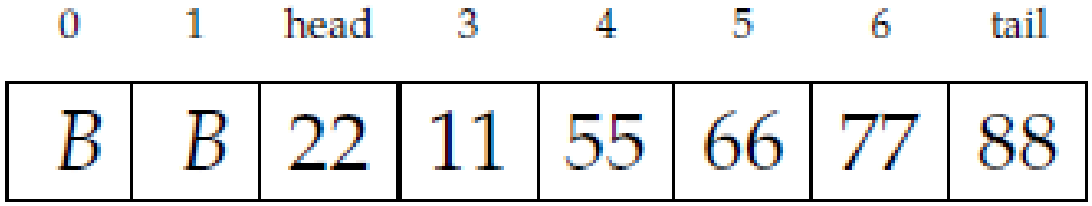
– dequeue();

– **enqueue(66); enqueue(77); enqueue(88);**

– dequeue();

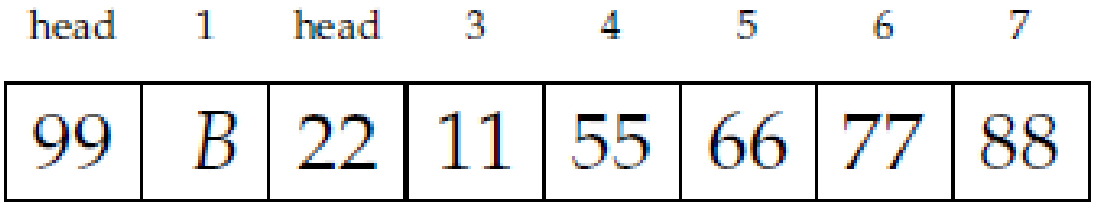
– enqueue(99);

Queues – Circular Arrays



– enqueue(55);
– dequeue();
– enqueue(66); enqueue(77); enqueue(88);
– **dequeue();**
– enqueue(99);

Queues – Circular Arrays



– enqueue(55);
– dequeue();
– enqueue(66); enqueue(77); enqueue(88);
– dequeue();
– **enqueue(99);**

Queues – Circular Arrays

- Let's look at sample code – queArrCir program.

Questions