COP 3502 - 9/14/23

(0) ULA Reminder
(0.1) P2 Picture

(1) Queues
(2) LL Implement
(3) Array Implementation
(4) Maths - Grid (Breadth First Search)

Queue - Line

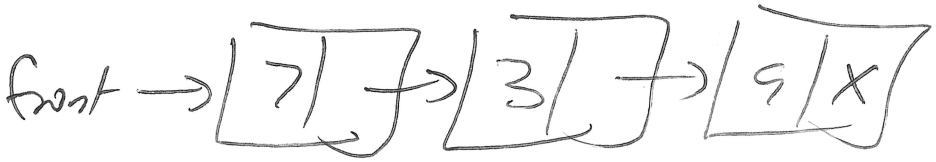
First In, First Out (FIFO)

O(1)
- enqueue - adds item to back of line
- dequeue - remove + return the front item from the queue

enqueue(7) → 7, 3, 9, 12
enqueue(3)
enqueue(9)
x = dequeue()
x [7]
enqueue(12)
y = dequeue()
y [3]

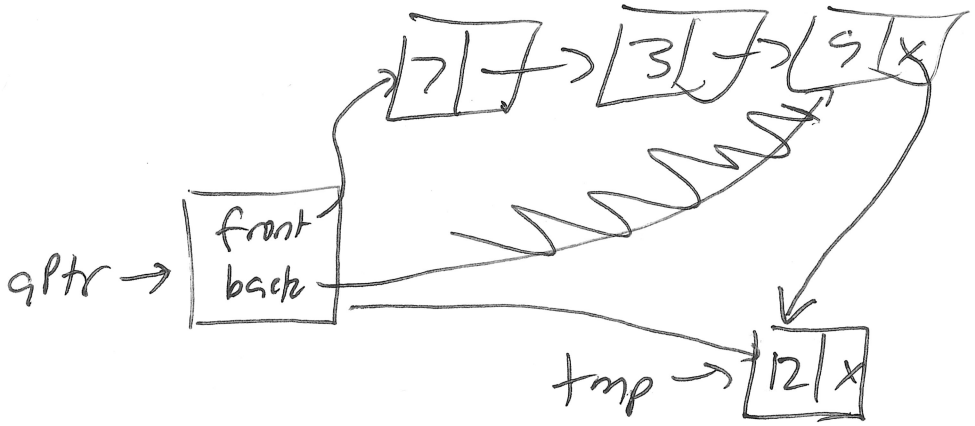
- front - return w/o removing front item
- size - returns # items has
- empty - returns true iff 0 elements
- full - can't add anymore.

Linked List



If I only have ptr to the front of the list, how long will it take me to enqueue something?

Ans: O(n), where n = # items in the list




```
typedef struct queue {
    node* front;
    node* back;
} queue;
```

enqueue: (1) create node

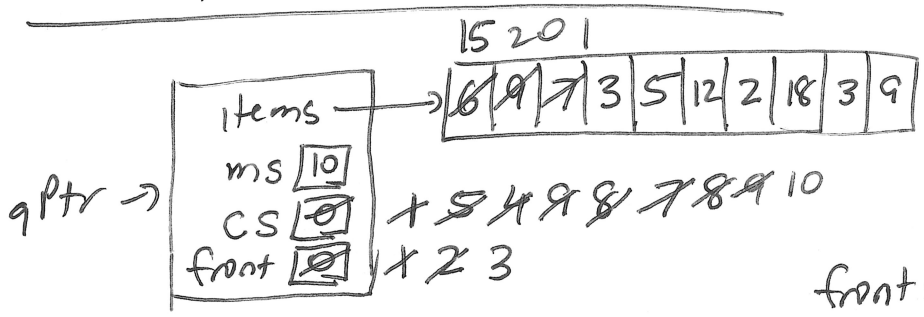
(2) link back to tmp

```
qPtr->back->next = tmp;
```

(3) Change back

```
qPtr->back = tmp;
```

Array Implementation



front + cs = 1st empt index

(qPtr → front + qPtr → cs) % %

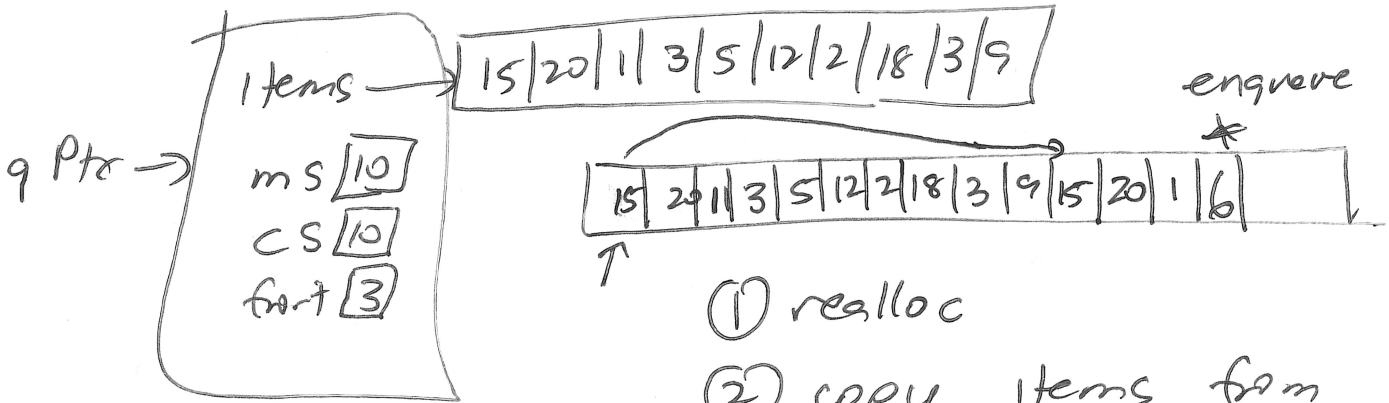
qPtr → ms

enqueue(6)
enqueue(9)
⋮
x = dequeue()
x 6
enqueues;
dequeues;
* enqueue(15)
enqueue(20)
enqueue(1)

Dequeue

```
qPtr->front++;
qPtr->front %= qPtr->ms;
```

Full queue

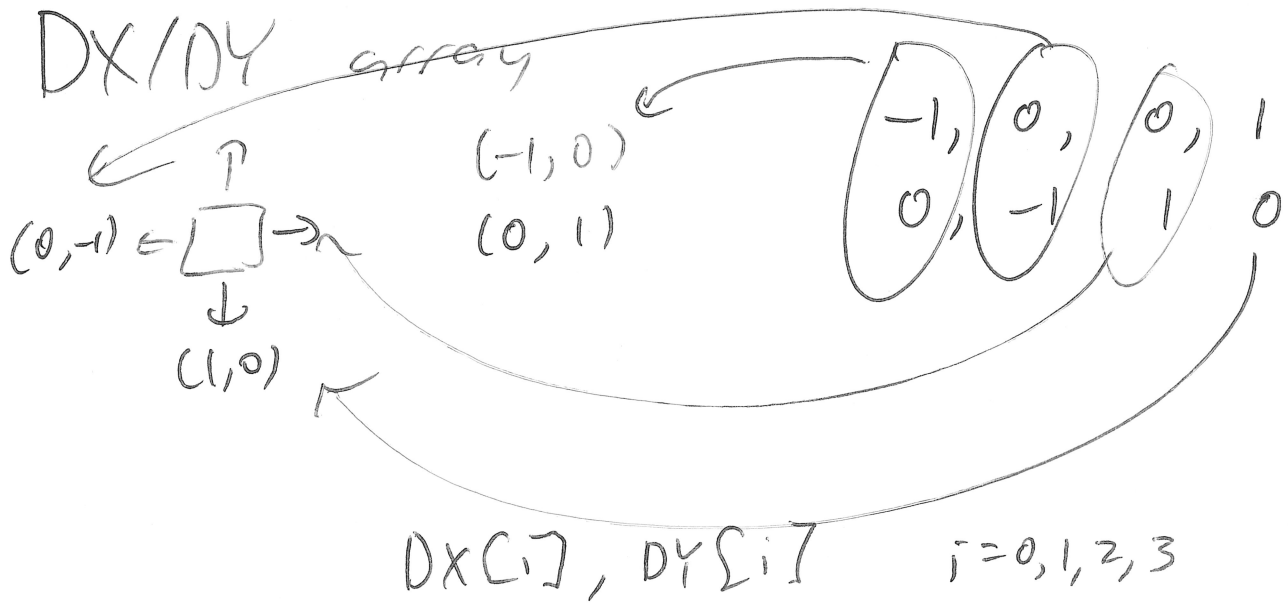


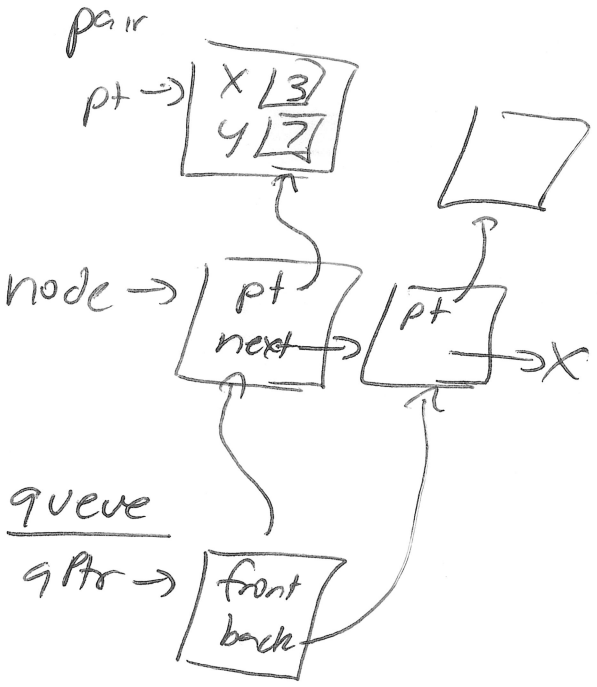
① realloc

② copy items from index 0 to index front-1 "to the back"

```
int n = qPtr->ms; //old
for(int i=0, j=n; i<qPtr->front; i++, j++) {
    qPtr->items[j] = qPtr->items[i];
}
```

enqueue(6)


DX/DY array
p
(0,-1)
(1,0)
(-1,0)
(0,1)
-1, 0, 0, 1
0, -1, 1, 0
DX[i], DY[i] i=0,1,2,3


pair
pt
x 3
y 7
node
pt
next
pt
x
queue
qPtr
front
back