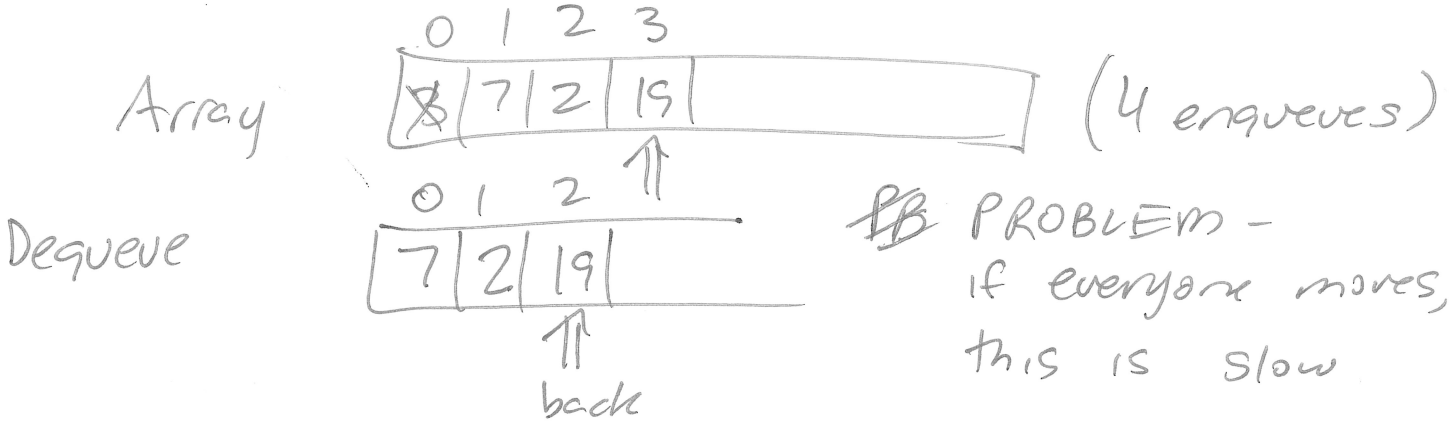
COP3502 2/14/22

~~Linked List~~ Implementation of a Queue
Array

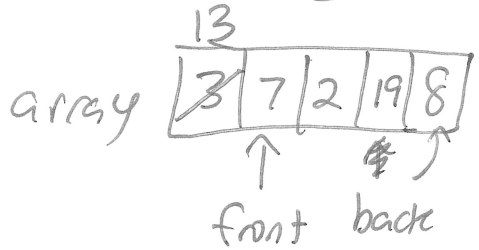
Queue - Breadth First Search
- Maze

If time, prints out a "heart" for V Day!



~~PB~~ PROBLEM -
if everyone moves, this is slow

ALTERNATIVE



enqueue - wrap around to front (% mod)

enq(3)
enq(7)
enq(2)
enq(19)
x=deq()
enq(8)
enq(13)

ISSUE

front + back ambiguous for empty vs. full

STORAGE

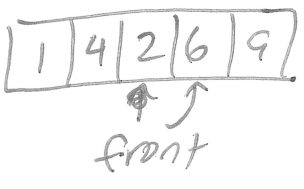
```
int front;
int size; // # items in queue
int length; // length of array
int* array;
```

ENQUEUE: (front + size) % length
↑ index to insert to

DEQUEUE: front

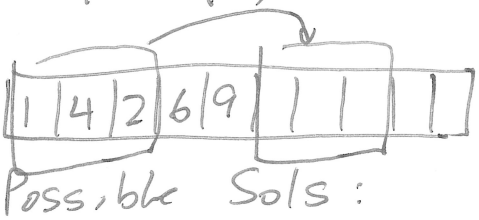
Maintain rest info to be consistent.

Doubling array size



front 3
size 5
length 5

Q: 6, 9, 1, 4, 2

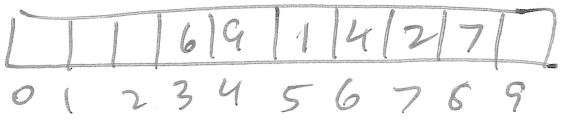


Where does ↑ go?

Two Possible Sols:

(1) copy everything back to index 0 as front

more efficient * (2) Copy indexes 0 to front-1 to end
keep front same



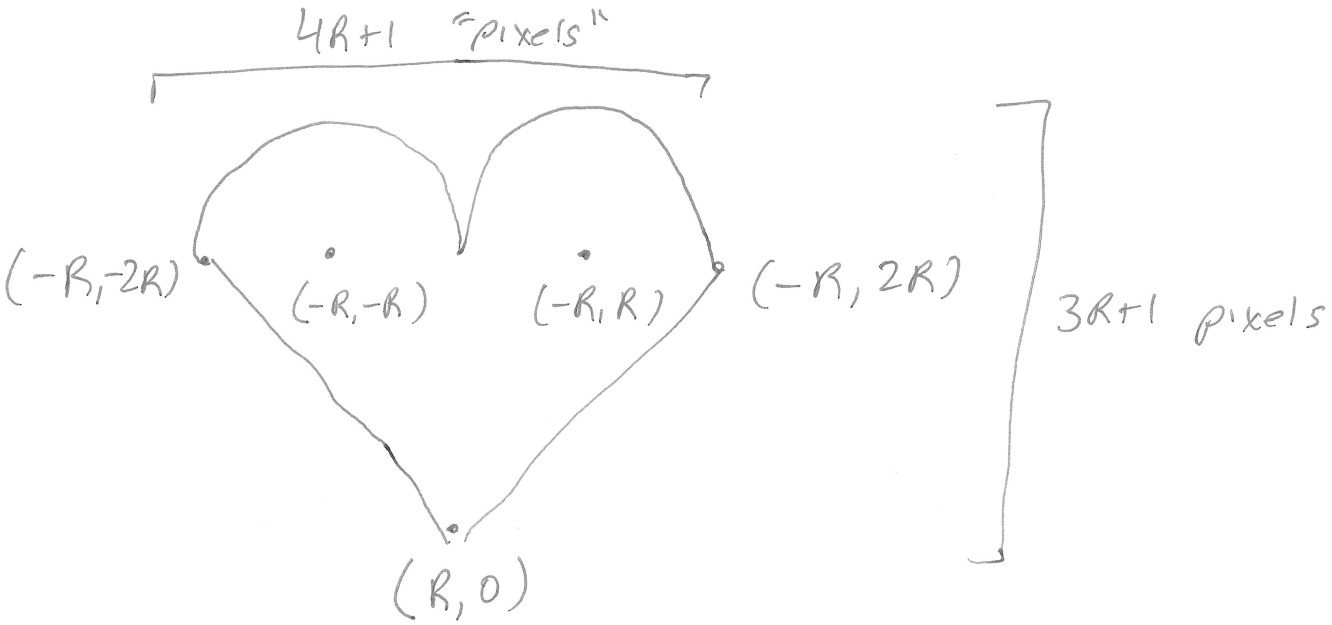
front 3
size 6
length 10

(3) Copy indexes front to old length to end of new array. Move front.

On a grid size r rows by c cols
location (x,y) [row x, col y $0 \le x < r, 0 \le y < c$]
maps to integer $x * c + y$.

To extract coordinates, $(x*c+y) / c \rightarrow x$
$(x*c+y) \% c \rightarrow y$

Heart Design #1



Heart Design #2

Squash circles into ellipses.
In distance squared calculation, mult DX by 2, so max I↕ is shorter.