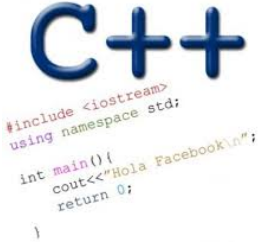
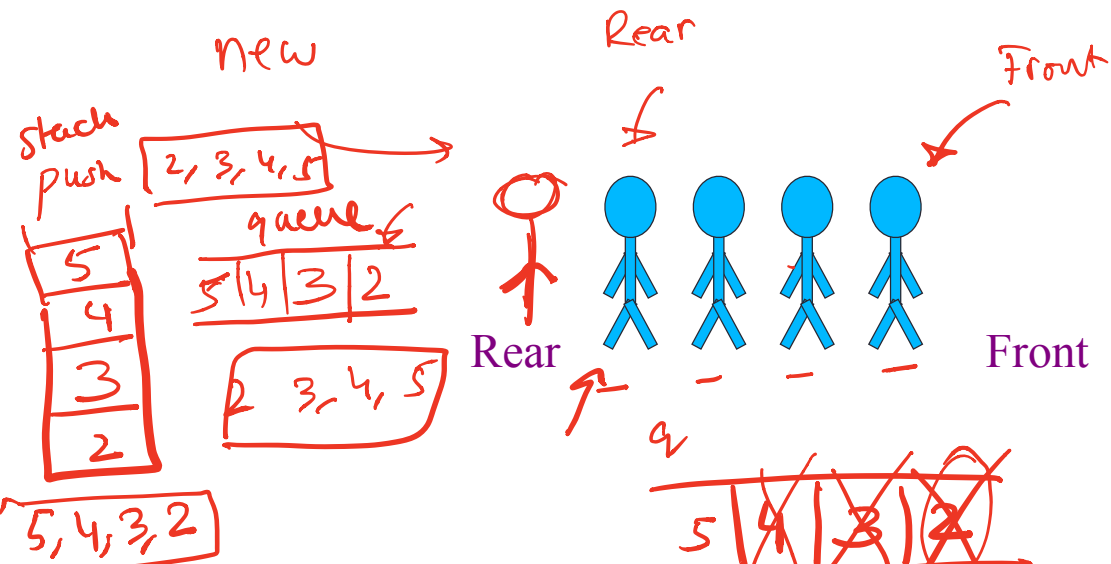
QUEUES
INTERVIEW PRACTICE

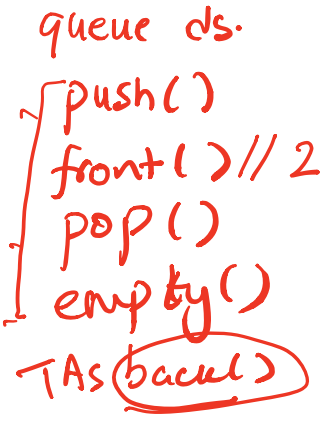
Problem Solving with Computers-II



The Queue Operations

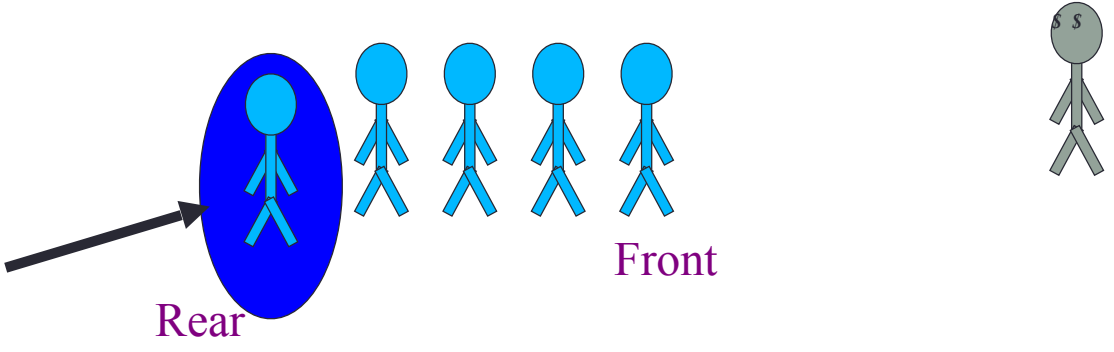
- A queue is like a line of people waiting for a bank teller.
- The queue has a **front** and a **rear**.





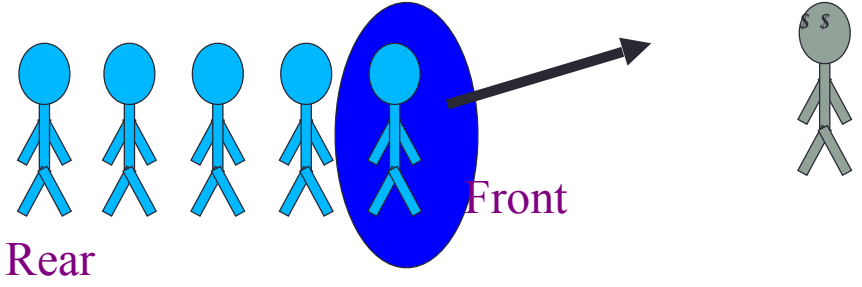
The Queue Operations

- New people must enter the queue at the rear. The C++ queue class calls this a **push**, although it is usually called an **enqueue** operation.



The Queue Operations

- When an item is taken from the queue, it always comes from the front. The C++ queue calls this a **pop**, although it is usually called a **dequeue** operation.



The Queue Class

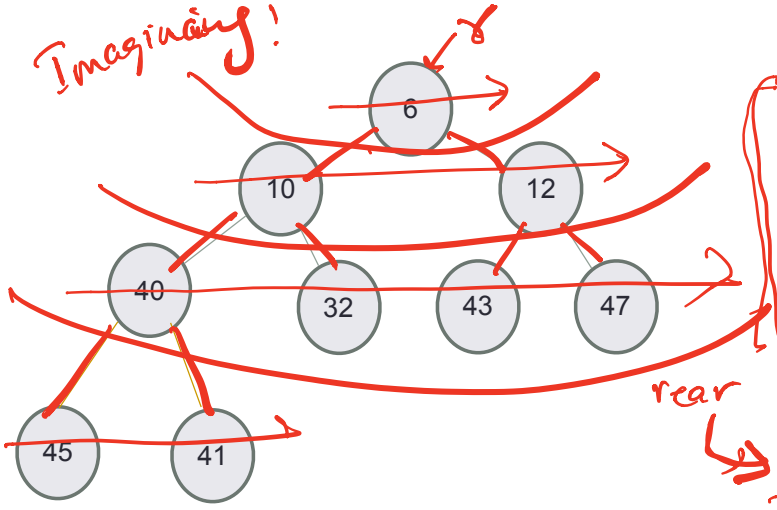
- The C++ standard template library has a queue template class.
- The template parameter is the type of the items that can be put in the queue.

```
template <class Item>
class queue<Item>
{
public:
        queue( );
        void push(const Item& entry);
        void pop( );
        bool empty( ) const;
        Item front( ) const;
        …
```

search

Breadth first traversal

Pre order post order in order



Breadth first traversal :

- Take an empty Queue.
- Start from the root, insert the root into the Queue.
- Now while Queue is not empty,
  - Extract the node from the Queue and insert all its children into the Queue. (pop())
  - Print the extracted node.

rear → queue → front

32 40 12 ~~10~~ ~~6~~

Output

| 6 | 10 | 12 | 40 | 32 | 43 | 47 | 45 | 41 |

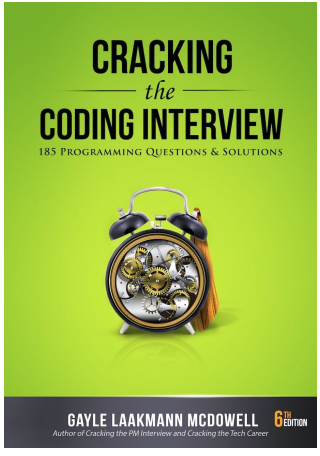
Vector

STL

```
#include <queue>

queue <int> q;
q.push(20);
q.front();
q.pop();
q.rear();
q.empty();
```

Small group exercise



Write a ADT called ir minStack that provides the following methods

- push() // inserts an element to the "top" of the minStack
- pop() // removes the last element that was pushed on the stack
- top () // returns the last element that was pushed on the stack
- min() // returns the minimum value of the elements stored so far
- empty()// returns true if minStack is empty

O(1)

A. I used this strategy
B. Something else

Stack <int> s;
int min;

S
2
4
3
10

hint
?
min: 10 3 2

How far did you get with this problem?

A. Code a correct solution & test it on some examples

B. Identified a strategy & tested it but didn't get the chance to code

C. Didn't reach any clear strategy

D. Didn't attempt

How useful did you find this exercis

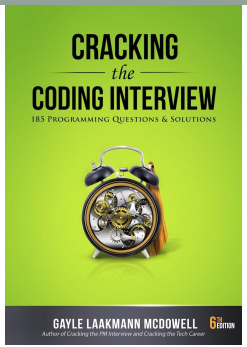
A. Very useful

B. Somewhat useful

C. not useful

Queue via stacks

Implement a MyQueue class which implements a queue using two stacks




```
class MyQueue {
public:
    push();
    pop();
    front();
    empty();

private:
    stack<int> s1;
    stack<int> s2;
};
```

Next lecture

* Wrap up