Stacks and Queues

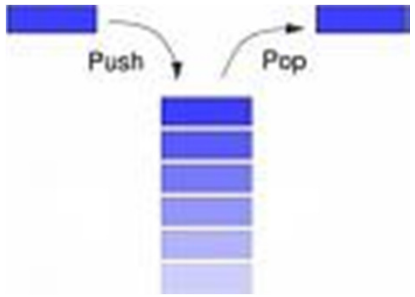
15-121

Introduction to Data Structures

Ananda Gunawardena

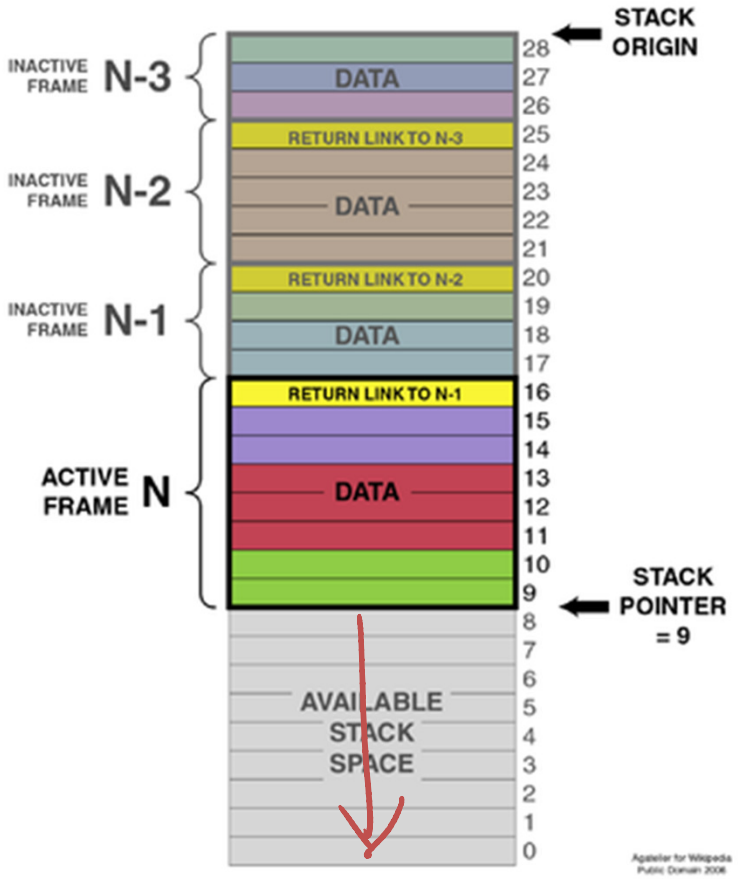
Stack and Heap

- A stack is a first-in-last-out structure



  - When a method is called computer stores all related data in a place called stack
  - When the method returns, stack is used generate the return parameters

Basic Architecture



- Stacks retain Information About where to return

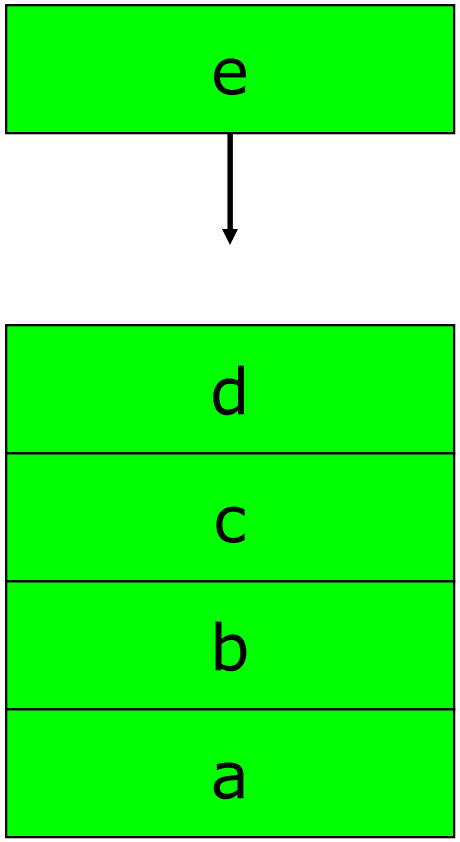
Stacks can be used (implicitly) to implement a technique called recursion

A Stack interface

```
public interface Stack {
  public void push(Object x);
  public void pop();
  public Object top();
  public boolean isEmpty();
  public void clear();
}
```

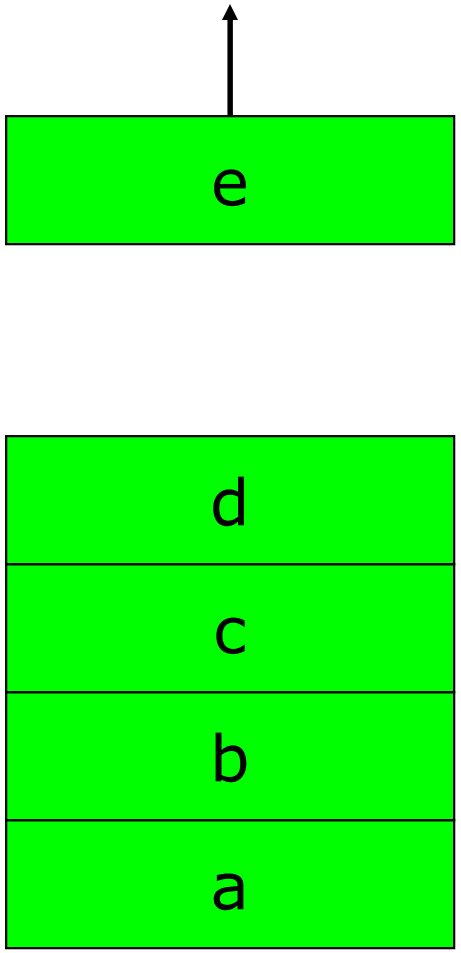
Stacks are LIFO

Push operations:



Stacks are LIFO

Pop operation:

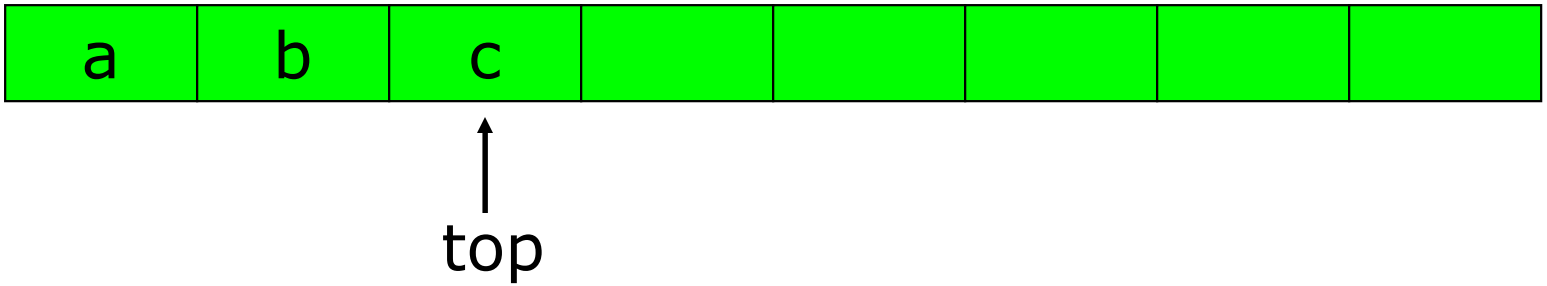


Last element that was pushed is the first to be popped.

Implementing Stacks

Implementing stacks using arrays

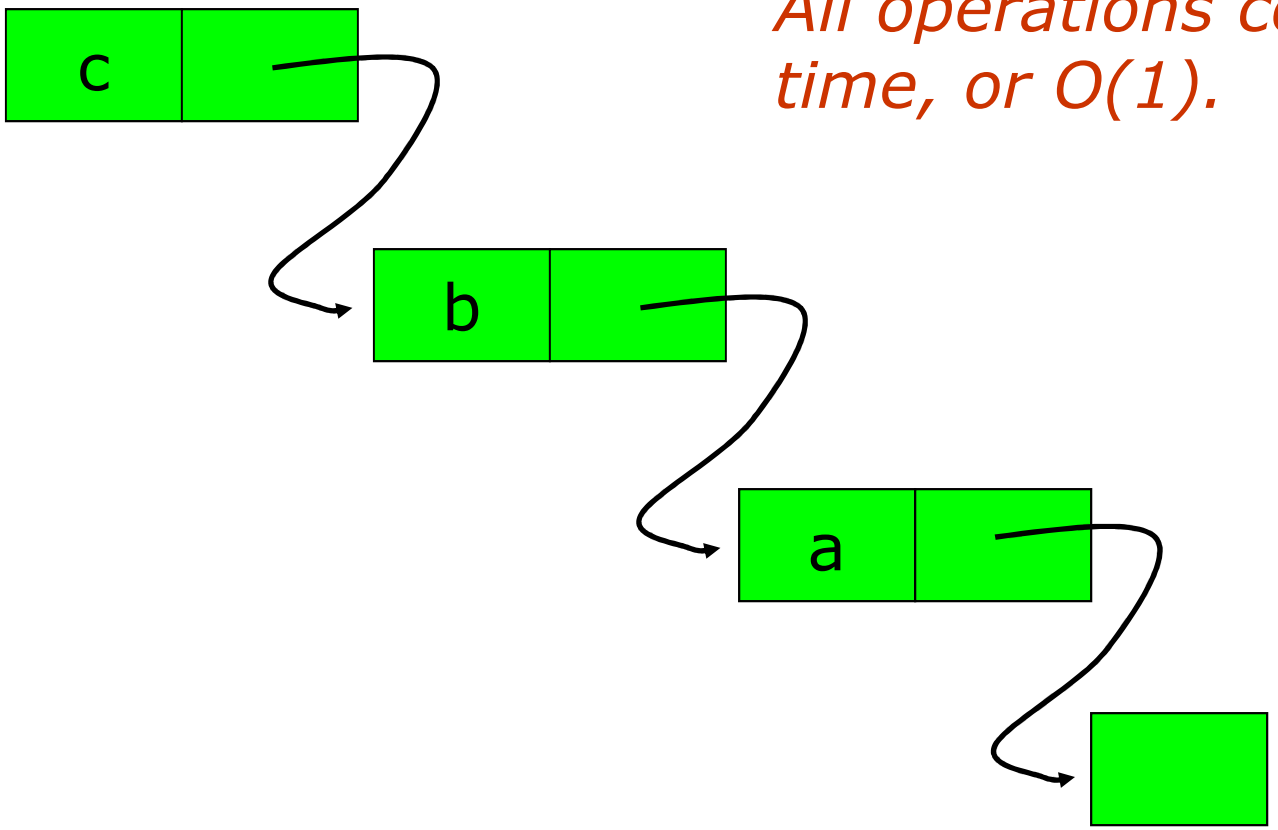
- Use an array-based representation.



- *What are some advantages and disadvantages of an array-based representation?*

Implementing stacks using linked lists

Linked representation. All operations constant time, or O(1).



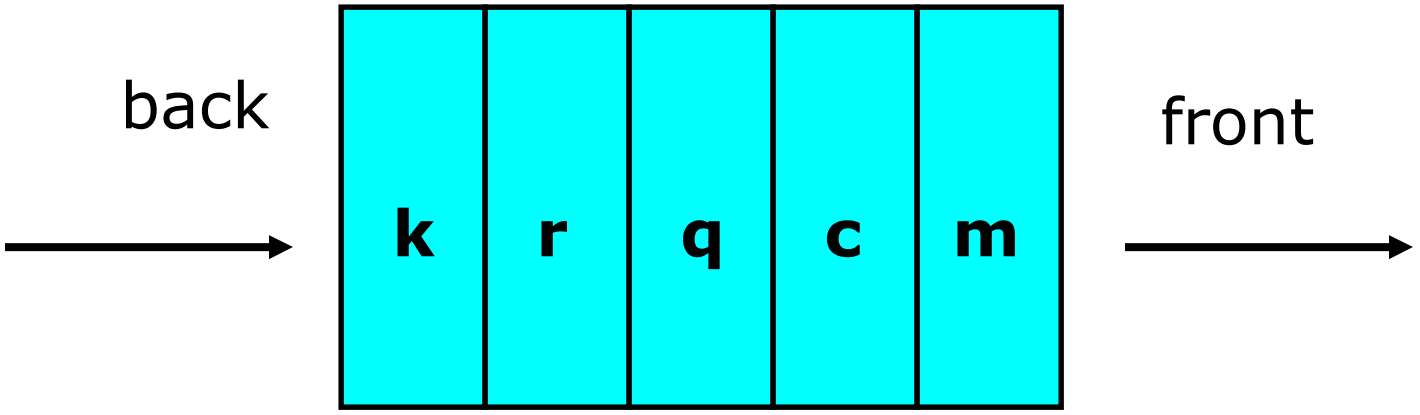
Queues

- Behavior
  - add item only from the back (enqueue)
  - remove items only from the front (dequeue)
- Many Applications
  - printer queue
  - network queues
  - common resource sharing
- Queue Operations
  - enqueue
  - dequeue
  - clear
  - empty
  - full

A Queue interface

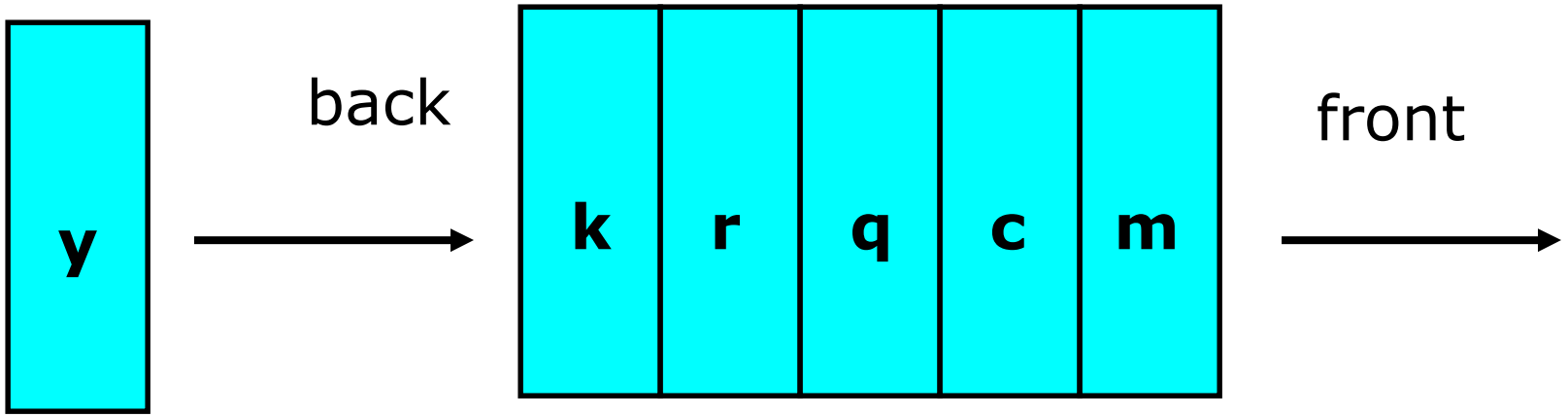
```
public interface Queue {
  public void enqueue(Object x);
  public Object dequeue();
  public boolean isEmpty();
  public void clear();
}
```

Queues are FIFO



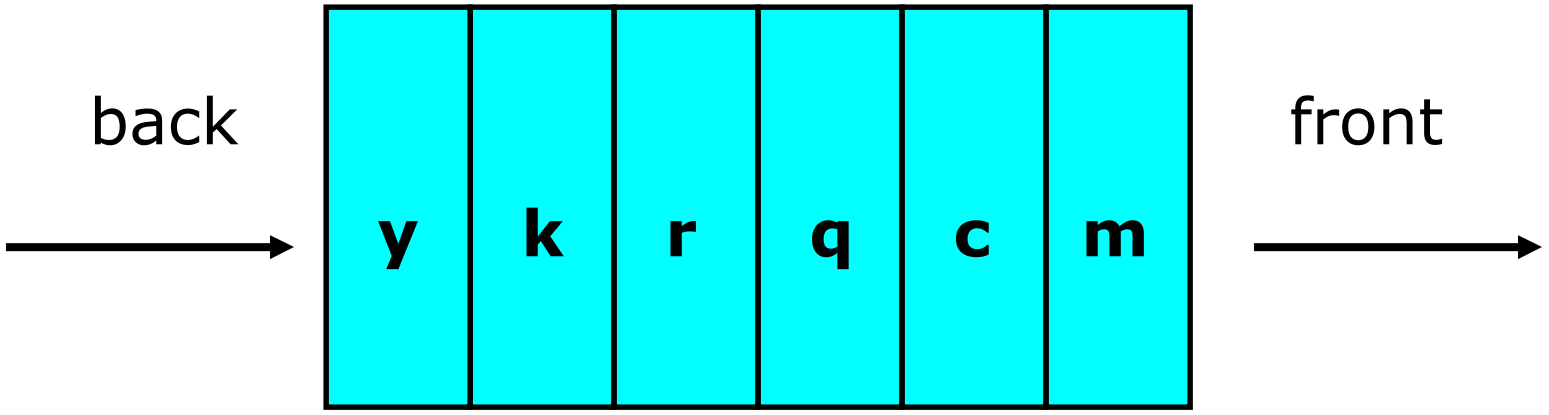
Queues are FIFO

Enqueue operation:



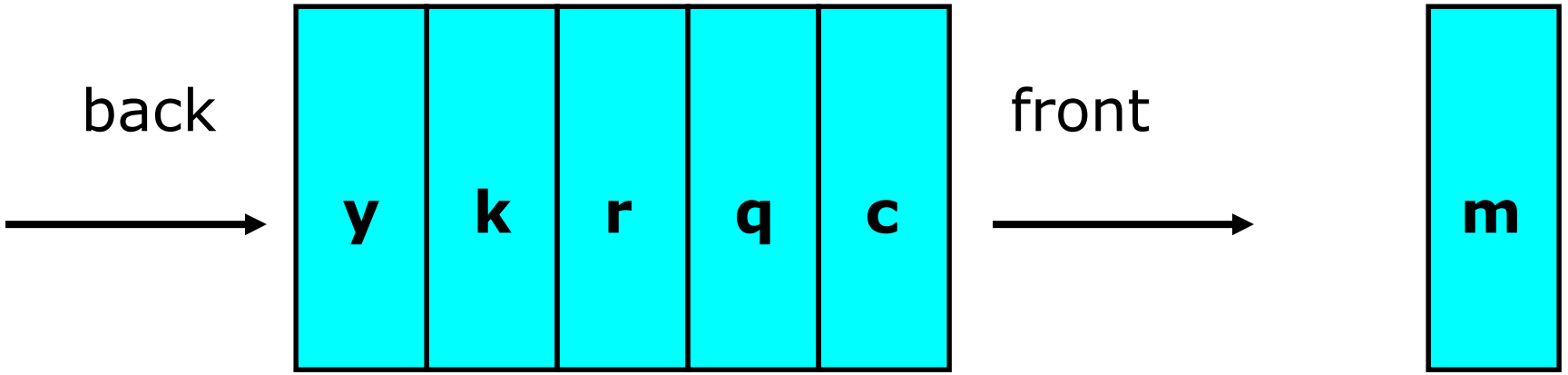
Queues are FIFO

Enqueue operation:



Queues are FIFO

Dequeue operation:



Implementing Queues

Implementing a Queue with an array

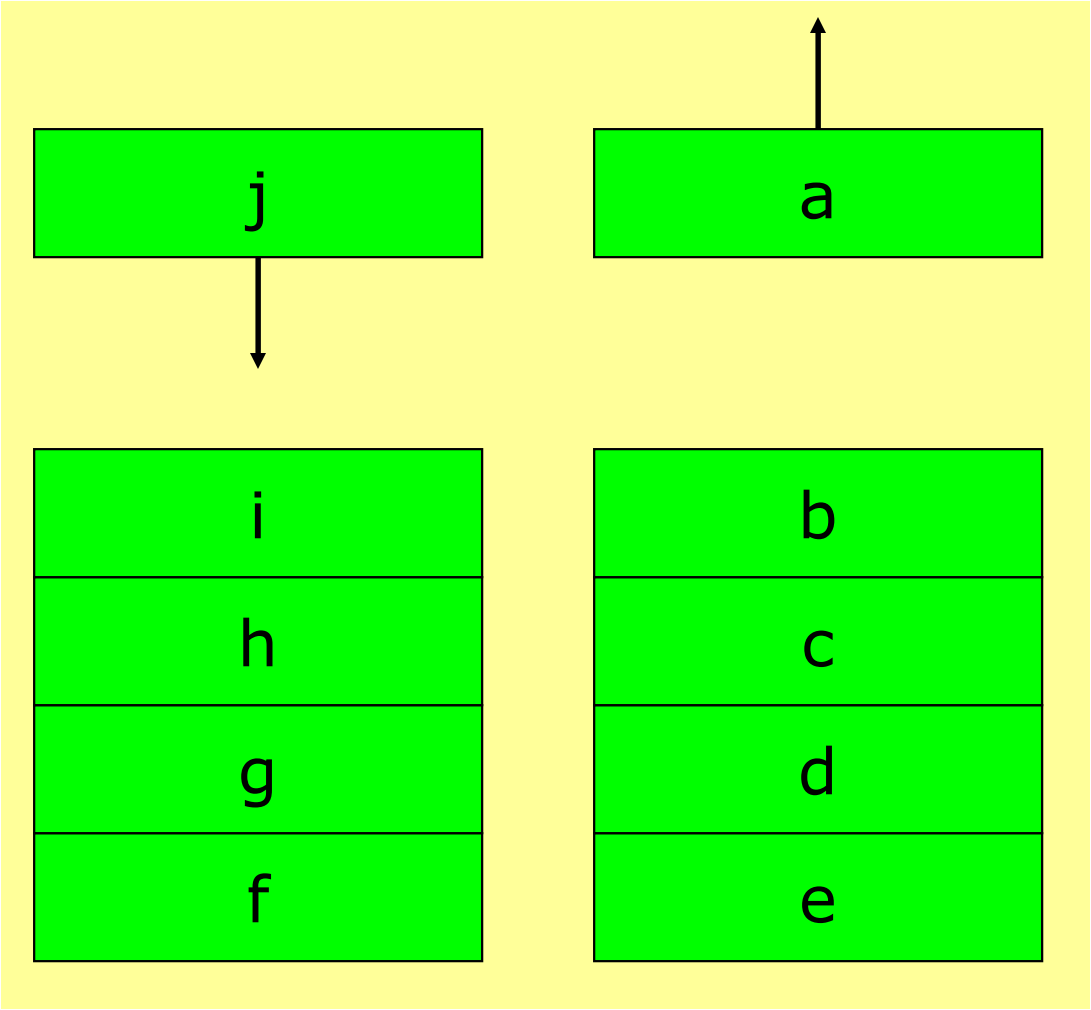
- What are the disadvantages?

Implementing a Queue with an circular array

- What are the advantages?

A queue from two stacks

Enqueue: **Dequeue**:



What happens when the stack on the right becomes empty?

Applications

Applications of Stacks

- Balancing Symbols
- Problem: Write a program that will check the validity of a statement
- Eg: (2 + 3 (4 – 5)) is valid
- (2 + 3 (4 – 5) is not valid

- Discuss an algorithm using stacks

Infix to postfix algorithm

- Read in the tokens one at a time
- If a token is an integer, write it into the output
- If a token is an operator, push it to the stack, if the stack is empty. If the stack is not empty, you pop entries with higher or equal priority and only then you push that token to the stack.
- If a token is a left parentheses '(', push it to the stack
- If a token is a right parentheses ')', you pop entries until you meet '('.
- When you finish reading the string, you pop up all tokens which are left there.
- Arithmetic precedence is in increasing order: '+', '-', '*', '/';

Applications of Stacks ctd..

- Evaluating Postfix expressions
- Consider the expression

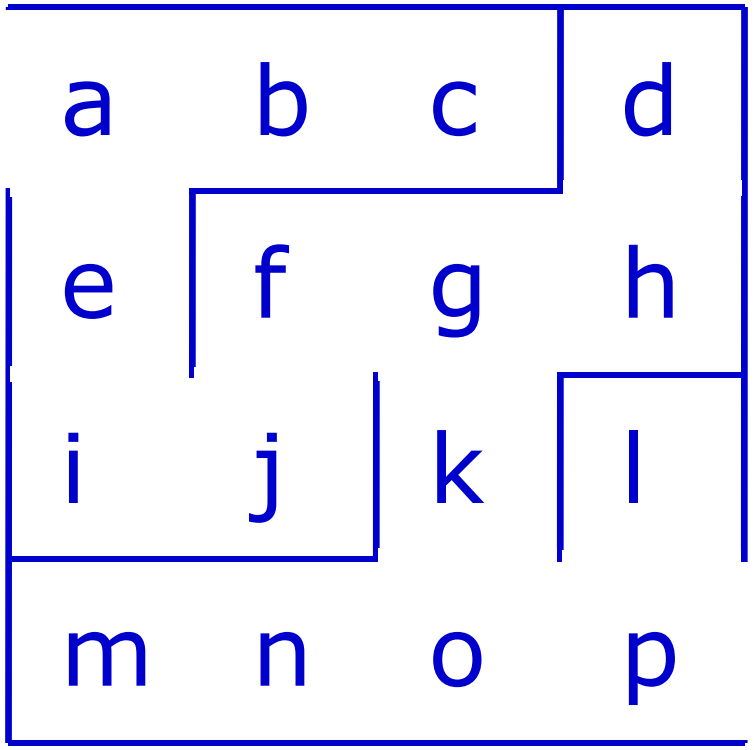
  (2+ 3*(5 – 2) + 3*2)
- How do we write a program to find the answer to the above expression?
- Write the expression in postfix notation

  2 3 5 2 - * + 3 2 * +

- Use a stack to evaluate it.

Stack Application - Maze

- Think about a grid of rooms separated by walls.
- Each room can be given a name.



Find a path thru the maze from a to p.

Applications of Queue

- Breadth First Traversal

Applications of Queue

- Snake Game

To Do

- Read notes on recursion
- Think of more applications where a stack or queue can be used
- Read the assignment on Mazes