Stacks and Queues

UNIVERSITY OF CALGARY

CPSC 319 - Data Structures

Benyamin Bashari

Example (Stack)

- push(2)
- push(1)
- pop() -> ?
- pop() -> ?
- push(25)
- pop() -> ?
- pop() -> ?
- push(17)
- pop() -> ?

Example (Queue)

- enqueue(2)
- enqueue(1)
- enqueue(25)
- dequeue() -> ?
- dequeue() -> ?
- enqueue(17)
- dequeue() -> ?
- dequeue() -> ?
- enqueue(14)
- enqueue(7)
- dequeue() -> ?

Stack Class in Java

- Constructor
  - Stack(): Creates and Empty Stack
- Methods
  - boolean empty(): Tests if this stack is empty.
  - E pick(): Looks at the object at the top of this stack without removing it from the stack.
  - E pop(): Removes the object at the top of this stack and returns that object as the value of this function.
  - E push(E item): Pushes an item onto the top of this stack.
  - int search(Object o): Returns the 1-based position where an object is on this stack.

Queue Interface in Java

- Methods
  - boolean add(E e): Insert if not violate capacity restrictions, true if success, exception if no space is available.
  - Boolean offer(E e): Same as add without exception.
  - E element(): Retrieves, but does not remove, the head of this queue.
  - E peek(): Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty.
  - E poll(): Retrieves and removes the head of this queue, or returns null if this queue is empty.
  - E remove(): Retrieves and removes the head of this queue.

Queue Interface in Java

- Implemented Class
  - LinkedList
    - Keep track of the elements with a simple linked list. The order of operations are O(1)
  - PriorityQueue
    - Keep track of the elements with more sophisticated data structures (could be also linked list but randomized). The order of enqueue() and dequeue() is O(log n). It can keep the elements sorted.

Postfix expressions

- We are used to infix expressions, such as
  - A + B * C
  - (A+B) * C
- Implementing a code that evaluates an infix expression is hard, because of parentheses and different priorities.
- Postfix expressions:
  - Binary operation is written after the two variables.
  - A + B -> A B +
  - A + B * C -> A B C * +
  - (A+B) * C -> A B + C *

Postfix expressions

- There are different ways to convert an infix expression to postfix expression.
- Evaluating a postfix expression is very easy.
- One of many applications of stack is to evaluate postfix expressions.

Example

- Infix: [(2 + 3) * 7] / (1+ 4), postfix: 2 3 + 7 * 1 4 + /
  - <top>
  - <top, 2>
  - <top, 2, 3>
  - <top, 5>
  - <top, 5, 7>
  - <top, 35>
  - <top, 35, 1>
  - <top, 35, 1, 4>
  - <top, 35, 5>
  - <top, 7>
  - Answer = 7

Algorithm

```
evaluate(postfix_expression):
    S = Stack()

    for elem in postfix_expression:
        if elem is number:
            S.push(elem)
        else:
            operation = elem
            //There must be at least two numbers in stack
            second_operand = S.pop()
            first_operand = S.pop()
            S.push(eval(first_operand, second_operand, operation))

    //Stack size must be exactly 1
    return S.pop()
```