8 – Stacks and Queues

Sunday, October 28, 2018 7:48 PM

Stacks

Stacks are data-structures that follow the LIFO (last-in-first-out) principle.

They support the following methods:
- Push – take an item and add it to the stack
- Pop – return and remove the last added item

There is also the "peek" method which gets the value of the last added item without modifying the stack, but this is not strictly part of the definition for a stack but if commonly included in implementations.

There are two obvious overflow errors:
- Stackoverflow – pushing to a "full" stack (using more memory than you were supposed to)
- Stackunderflow – popping from an empty stack

A stack is used in the execution of programs. Functions are pushed to the stack and they can call themselves (**recursively**) or call other functions that then also get pushed to the stack. When a function completes, the program continues from the pointer to the memory address of the previous function it was called from which gets pushed to the stack with the function (along with parameters and room for local variables.

Queues

Queues are data-structures that follow the FIFO (first-in-first-out) principle.

They support the following methods:
- Enqueue – add an item to the queue
- Dequeue – return and remove the oldest element (first) in the stack

For instance, the computer's serial buffer is a queue. Bytes that arrive on the serial bus are "queued", and then when you read from the serial line, you "dequeue" the bytes in the order that they were received.