CS280 Notes 2/8

by Christian Sagel

When returning a char pointer to the board, do we have to make a copy of whatever struct we have and pass it? Yeah, that's okay.

Infix (with parens) to Postfix

5 * (((9+8) * (4*6)) + 7) = 2075

Operands: Send to the output
Left parenthesis: push onto stack
RIght parenthesis: operators are popped off the stack and sent to the output until a left parenthesis is found (and then discarded)
Operator: If the stack is empty, push the operator.

We can do the operation with a stack class.

For a linked list we want to push back/front because its constant for doubly-linked lists.
Array-based containers are cache friendly, close to each other in space.

Queues: Similar to stacks but more general. Usually mean a FIFO queue (First-In, First-Out). Add an item to the front and remove an item from the back.
There are other policies for other queues: Add to either end, remove from either end.

Implementing a FIFO queue using a circular array
We can't assume the array is indexed from 0 to Size 0 1.
These kind of data structures are heavily used in the OS where every cycle counts.

Deque: A double-ended queue. They allocate an array.
Queues and stacks can be implemented using arrays or linked lists. They are temporal data structures because they reflect when something was added.
There are standard classes in the STL

PriorityQueue: push something to it, but always pull out the largest value.
The complexity o.f the algorithms depends on how the list/array is implemented. (Sorted vs unsorted)
Sometimes it's okay to spend time upfront and optimize speed for later.

4 different factors: Arrays, Linked-lists, Sorted, Unsorted

In a priority queue we can rearrange and move the element in the back to fill in the index where an element was removed, because what we want to do is just validate the data structure again. Because its unsorted, we can do whatever we want.
PriorityQueues will be very useful for a later Graph assignment.
The tree list is node-based, though its performance characteristic comes from how the nodes are linked up.

Most, if not all data structures are either array-based or node-based.
We will spend a good 6 lectures on trees.

Trees

They are on the fundamental data structures. Balanced tree to sorted linked list.
The nice thing about balanced lists is about being able to do a binary search.
Doing trees recursively will be way easier than iteratively.
There are many kinds of trees, mostly node based.
Trees are a very specific type of graph.
Typical search times for trees are O(log N) (

In the trees, we call them nodes or vertices (what carries the information).
Edges are connections between two vertices.
The root of a tree is analogous to the head of the linked list.
A recursive data structure. Every node itself is a subtree.
Nodes can only have one parent. The exception is the root with no parent.
Relationships between siblings, parents, uncles, etc.
Paths: A list of vertices. From the root node to any one node there's exactly one path to each node. No duplicate nodes.
Leaf: A node with no children.

A linked list has the same height and length.
To find the height of a tree is a linear operation, just like to find the width.
With trees it is possible to end up stuck

Binary trees called so for only one or two children per node.
The level of a given node in a tree is defined recursively:
0 if node is root
Level(parent) + 1 if node is child of parent

If the height between two nodes is greater than 0, the tree is unbalanced.
A complete binary tree is similar to a balanced binary tree except that all of the leaves must be placed as far to the left as possible.

Traversing Binary Trees

Recursive algorithms:

- Pre-order traversal: Visit, traverse left, traverse right. The root will be the first node visited.
- In-order: traverse left, visit , traverse right
- Post-order traversal: traverse left, traverse right, visit Last node visited root.

For a BST, we will do in in-order traversal.

Regarding second tree example:

Pre - GDBACFEFKHJIML
In - ABCDEFGHIJKLM
Post - ACBFEDIJHLMKG

Node H is unbalanced because the difference in height between the lowest left and right children.

Expression trees: Compilers do these kind of folding operations with expression trees to evaluate.