Lecture 14 Notes

Derek Harter

2015-05-06

1 First Session (11 - 11:40)

1.1 Stacks

- A stack is a structure that allows nodes to be added and removed from the top of the stack only.
- It is last-in-first-out (LIFO) data structure. The last item you push, will be the first item back out when you pop.
- One way to implement, is using a linked list, where we provide only 2 operations, ‘push()˜ and `pop()`
- Function push() is basically the same as the `insertAtFront()` function we have already developed.
- Function pop() removes item from top
- Other common operators on a stack are a function to test if the stack is empty.

```
struct Node
{
  int data;
  Node* nextPtr;
};
```

2 Second Session (11:45 - 12:30)

2.1 Queues

- A queue is basically similar to a line in a store. The first person in line is serviced first.

- Queue nodes are added to the tail of the queue, and queue nodes are removed from the head of the queue.
- A queue is first-in, first-out (FIFO), the first item in will be the first item removed.
- The basic queue operations are usually named `enqueue()` and `dequeue()`
- Function `enqueue()` adds a node to the tail of the list.
- Function `dequeue()` removes the node from the head of the list.
- For efficiency, we usually keep pointers to the head and tail nodes of a queue, so we can do `enqueue~/~dequeue` operations in O(1) constant time
- We also commonly provide a function that determines if the queue is empty or not.

3 Third Session (12:40 - 1:40)

3.1 Trees

- Linked lists, stacks and queues are linear data structures.
- A tree is nonlinear.
- Each node in a tree is either an internal node or a leaf node.
- All internal nodes have 1 or more children. In a strict binary tree (which we will discuss here) each internal node has 1 or 2 child nodes, the left child and the right child.
- In a tree, including a binary tree, there is one node that is at the top of the tree, known as the root node of the tree.
- For example, we can recreate our binary search by organizing items as a binary search tree.
  - The left node of tree contains child nodes with values less than the parent node.
  - The right node of tree contains child nodes with values greater than the parent node.