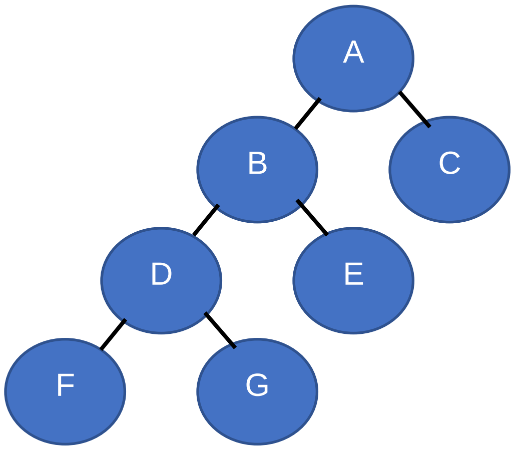
CS32 Final Study Guide

Firstly, know how to write an assignment operator, copy constructor and destructor for a simple class that has dynamically allocated memory.

Trees

1. Define the following terms
   - binary search tree
   - leaf of a tree
   - degree of a tree node
   - degree of a tree
   - path in a tree
   - length of a path in a tree
   - descendents of a node in a tree
   - ancestors of a node in a tree
   - depth of a node in a tree
   - height of a node in a tree
   - height of a tree
2. Draw the expression tree for the expression a*(b+c)-(d+(e-f)*g)
3. For the tree



   a. What is the height of node A?
   b. What is the height of the tree?
   c. What is the depth of node E?
   d. What is the parent of node F?
   e. Name the descendent(s) of node B.
   f. Name the leaves in the tree.
4. Write a C++ function `TreeNode* copy_tree(TreeNode* T)` that returns a copy of binary tree `T`

5. Write a C++ function `TreeNode* expand_leaf(TreeNode* node, ItemType x, ItemType y)` that returns a new binary tree that is identical to the binary tree T except that every leaf in `T` now has a left child and a right child whose values are equal to `x` and `y`, respectively. For example, invoking `expand_leaf(T, 9, 12)` on the tree on the left produces the tree on the right.

```
  5                   5
 /                   /
3                   3
                   / \
                  9  12
```

6. Write a C++ function `int height(TreeNode* T)` that returns the height of the binary tree `T`
7. Write a C++ function `bool same_tree(TreeNode* T1, TreeNode* T2)` that returns true if binary trees `T1` and `T2` are exactly the same (same values, same structure) and returns false otherwise. You may assume that values of `ItemType` can be compared by means of the == operator.
8. Write a C++ function `TreeNode* bst_insert(ItemType item, TreeNode* T)` that inserts `item` into the binary search tree `T`. After the insertion, `T` must remain a binary search tree. You may assume that `ItemType` values are comparable using the ==, <, and > C operators. Draw the binary search tree resulting from sequentially calling `bst_insert` on an initially empty binary tree with the values 9,12,5,15,7,10,18.

Tree Traversal

1. For the tree

```
  +
 / \
A  *
  / \
 B  –
   / \
  C   D
```

   a. Write the node labels in pre-order order.
   b. Write the node labels in in-order order.
   c. Write the node labels in post-order order.

Searching

1. Given an array containing the sequence `1,5,29,45,67,76,92,104,187,234` (in that order)
    a. State each comparison made in finding the number 234 using linear search. (For example, 234:1 is a comparison of 234 with 1.)
    b. State each comparison made in determining that the number 48 is not present using linear search.
2. Given the following sequence of integers (for example, integers received from the keyboard)

    `29,234,1,45,92,76,67,187,104,5`

    (in that order)
    a. Draw the binary search tree that would result from inserting the integers in the order given.
    b. State each comparison made in finding the number 234 using binary search. (For example, 234:1 is a comparison of 234 with 1.)
    c. State each comparison made in determining that the number 48 is not present using binary search.
    d. Write the integers as they would be encountered in an in-order traversal of the tree.

Sorting

1. How many permutations of N distinct items are there?
2. Explain why selection sort is $O(N^2)$ on average.
3. Explain why quicksort is O(N log N) on average, but $O(N^2)$ worst case.
4. Given an array containing the integers `12 9 3 6 4 1` (in that order)
    a. Show the array as it is sorted by the selection sort
    b. Show the array as it is sorted by the mergesort
    c. Show the array as it is sorted by the insertion sort
    d. Show the array as it is sorted by the bubble sort
5. Write a templated version of both bubble and selection sort.
    a. As the numbers of elements to be sorted increases how do the two sorts compare to one another?

Asymptotic Analysis

1. Put in increasing order based on the following set: $O(n^3)$, $O(2^n)$, O(n), O(n log n), O(log n), O(1), $O(n^2)$

2. Linear search can be done on data in an array or in a list. What is the average time complexity for successful search using an array? Using a list? Name an advantage of the list implementation. Name an advantage of the array implementation.
3. Binary search can be done on data in a sorted array or in a binary search tree. What is the average time complexity for successful search using the array? Using the BST? What is the worst case time complexity for each? Name an advantage of the BST implementation. Name an advantage of the sorted array implementation.

Hash Tables

1. Identify the differences between open and closed hashing
2. Given the following set `12 9 3 6 4 1` (in that order), in a Closed – Linear Probing hash table, if we apply the hash function h(x) = value % 11, where would 1 be placed?

Heaps

1. Be able to identify a heap.
2. Know what heapsort is, including its characteristics of time complexity.
3. Show a maxheap after inserting each of the following numbers: `13,9,4,5,2,8,3,0,12`
4. Remove the two biggest numbers, showing the heap structure after each deletion.