CHAPTER 5

Section 5.3

Section 5.3: Decision Trees

- Recall decision trees used for counting problems earlier.
- Example: Toss a coin 4 times. How many possible outcomes are there if we do not have 2 heads in a row?
- Formal Definition of a **decision tree**
  - A tree in which
    - each internal node represents an action (a coin toss).
    - arcs to children represent outcomes of that action (heads or tails).
    - leaves of the tree represent the final outcomes.

Decision Trees

- Decision trees can be useful for examining the actions of an algorithm.
- We will consider the following algorithms:
  - Sequential Search
  - Binary Search
  - Binary tree search

Sequential Search

- Searching a list for element x.
- Sequential search on unordered list L
  - Start at the first element in the list (i=1).
  - Repeat the following until either the element has been found, or the entire list has been searched.
    - Compare the current element in the list, $L[i]$, to x.
    - If $L[i] = x$, then stop.
    - If $L[i] \neq x$, then move to the next element in the list (i=i+1).

Example 31

- Draw a decision tree for sequential search on a list of 5 elements.
  - How many comparisons are done if x = L[3]?
    - What is the worst case scenario?
    - How many comparisons are done in the worst case?
  - How many comparisons for sequential search on a list with n elements in the worst case?

Binary Search

- Searching an *ordered* list for element x.
- Start at the middle element of the list.
- Repeat the following until there are no more elements to search
  - Compare node i to x
    - If $L[i] = x$, stop (x has been found)
    - If $x < L[i]$, do a binary search on the left half of the list.
    - If $x > L[i]$, do a binary search on the right half of the list.

Binary Search

- Example 32
  - Draw a decision tree for the binary search algorithm acting on a sorted list of eight elements.
  - How many comparisons are done in the worst case?
  - What is the height of this tree in relation to the number of elements in the list?

Lower Bounds on Searching

- Recall the minimum height of a binary tree with n nodes is $\lfloor \log_2 n \rfloor$.
- What is the minimum height of a decision tree for any searching algorithm if the list contains n elements?
- For worst case, the minimum number of comparisons will be equal to the height of the decision tree.

Binary Tree Search

- Data can be organized into a **binary search tree**.
- **Binary tree search** algorithm can then be applied to the binary search tree.

Binary Search Tree

- To build a binary search tree
  - First data item is made the root of the tree.
  - Compare successive items to existing nodes, beginning with the root.
    - If new item < node, test the node's left child.
    - If new item > node, test the node's right child.
    - If no child node, the new item becomes the child.

Binary Search Tree

- Example 33
  - Organize the following data items into a binary search tree:
    5, 8, 2, 12, 10, 14, 9

- Property of binary search tree
  - The value at each node is
    - greater than all the values in its left subtree, and
    - less than all the values in its right subtree.

Binary Tree Search

- Binary Tree Search algorithm can be applied to the binary search tree.
  - Compare item x with a succession of nodes beginning with the root
  - If x = node value, stop (x has been found).
  - If x < node value, check left child next.
  - If x > node value, check right child next.
  - If no child exists, stop (x is not in list)

Binary Search Tree

- Given the previous binary search tree
  - What is the depth of the search tree?
  - How many comparisons are done if $x = 12$?
  - How many comparisons if $x = 11$?
  - For a tree of depth d, how many comparisons would be done in the worst case?

Binary Search Tree

- Different binary search trees can be built for the same set of data values
  - Depends on the order in which the data items are inserted.
  - The depth of the tree can vary.

- Example 34
  - Build a binary search tree for the data items in the given order:
    9, 12, 10, 5, 8, 2, 14

Binary Tree Search

- For the previous example
  - What is the depth of the tree?
  - How many comparisons are done if $x = 12$?
  - How many comparisons are done if $x = 11$?
- The algorithm for building the tree can be adjusted to result in a more “balanced” tree.
- In what order must data items be entered to construct a worst case binary search tree?
  - A tree that would require searching all data items if the item is not in the list?