CSI 35 LECTURE NOTES (Ojakian)

Topic 16: Tree Applications

OUTLINE

(References: Rosen: 11.2)

1. Binary Search Trees
2. Decision Trees
3. Game Trees

---

1. Binary search trees

   Definition: For any vertex, all the vertices in its left subtree are smaller, while all the vertices in its right subtree are larger.

   PROBLEM 1. *Create a binary search tree from a list of numbers:*

   $$5, 2, 9, 7, 1, 4, 12, 6, 20.$$

   Then search for a number in the tree. Compare the search time in the tree versus the list.

2. Decision Trees

   (a) Create a decision tree for sorting 2 elements
   (b) Example 4: Create a decision tree for sorting 3 distinct elements.

3. Games

   (a) Example Games: Nim, Sliding (Nim in disguise), Chomp, Cram.
   (b) Nim and Chomp - can play online.
   (c) Our focus: 2-player, no-chance, win-lose-or-draw, perfect information (many also impartial games)
   (d) **Theorem**: For such games, exactly one player has a winning strategy, or both players have a draw strategy.
      i. Example: In Tic-Tac-Toe, both players have a draw strategy.
      ii. Checkers? Huge computer-aided work solved it!
      iii. Chess? No one knows ...

4. Game Trees

   (a) Example 7 (and Figure 8). Tic-tac-toe. Make a tree to describe game
   (b) Example 6 and 8 (and Figure 7 and 9). Misere Nim (Or: must leave something). Do minmax calculation to see who wins. Players can deduce strategies from that.

5. Exercises

   (a) Section 11.2. Binary Search: 1 - 5
   (b) Section 11.2. Games: 33 - 43 (not 35)
   (c) Section 5.2. Games and Induction: 11, 15, 16; include Chomp (Strategy-Stealing approach).