Problem Set No. 2

COS 521 Due: 10/17/02
Fall 2002
Tarjan

1. (Turns) A *left turn* is an application of the zig-zig step in splaying: do two left rotations, the first above the second. Prove: any sequence of intermixed left turns and left rotations on an n-node tree contains O(nlogn) left turns and $O(n^2)$ left rotations.

   Hints: Break this into two parts. First note that every left turn consists of two left rotations, and prove an $O(n^2)$ bound on left rotations. second, count turns by using a modification of the potential function useful to analyze splaying, with the following properties: (a) every left rotation decreases the potential; (b) every left turn decreases the potential by the least 1; (c) the difference between the maximum and minimum possible potentials (of an n-node tree) is O(nlogn).

   Open problem 6: Prove or disprove: any sequence of left turns on an n-node tree contains O(n) turns.

2. (Filtering Search) Describe a good data structure to store a set of intervals $[x_i,y_i]$, each with a real-valued priority z_i, that supports the following operations:

   find max(a): given a point a, return an interval $[x_i,y_i]$ such that $x_i \le a \le y_i$ and z_i is maximum.

   insert (x_i,y_i,z_i): insert a new interval $[x_i,y_i]$ with priority z_i.

   delete (x_i,y_i): delete internal $[x_i,y_i]$.

   Analyze the efficiency of your data structure. Your goal is to minimize the time per operation and the space needed.

   Open Problem 7: Devise a linear-space, logarithmic-time solution.