Problem Set No.4 - Corrected

COS 521 Due: Thursday, Dec 5, 2002
Fall 2002
Tarjan

1. (a) Consider the following operations on a forest of dynamic trees, each node of which has a real-valued weight:

   find weight(v): return the weight of node v.

   find total(v): return the total weight of all descendants of v, including v.

   add all(v, Δ): add Δ to the weight of all descendants of v, including v.

   link(v,w): make v a child of w(v must be a root).

   cut(v): make v a root, removing it as a child of its parent if it has a parent.

   Describe an implementation of these operations with an amortized time bound of O(log$n$) per operation on a tree or trees with n nodes.

   (b) Consider the set of tree operations as above, but with *find total* replaced by the following operation:

   find max(v): return a descendant of v of maximum weight.

   Describe an implementation of the modified set of operations with an amortized time bound of O(log$n$) per operation on a tree or trees with n nodes.

   Note: the hard part of this problem is in handling high-degree nodes.

2. Consider a sequence of real numbers $v_1,v_2,\ldots,v_n$. The sequence is *deque-sortable* if it can be sorted by a sequence of the following operations using a single deque d, initially empty. The initial input is $v_1,v_2,\ldots,v_n$.

   push: remove the next element from the input and push it onto the top of d.

   pop: remove the top element on d and add it to the output.

   inject: remove the next element from the input and inject it into the bottom of d.

   eject: remove the bottom element from d and add it to the output.

   (a) Give an example of a sequence that is not deque-sortable, and prove it.
   (b) Describe and analyze the fastest algorithm you can for testing whether a given sequence is deque-sortable. (There is an O(n)-time solution.)