Scott Fenton

Homework 3

CS310

1. Can all of the following be supported in logarithmic time: insert, deleteMin, deleteMax, findMin, and FindMax?

Yes, with a binary search tree with rebalancing, such as a AVL tree will avoid worst case performance of O(n). Each of the following methods can be supported in logarithmic time. You can search, delete and insert in O(logn) time.

1. Show that the following operation can be supported in constant time simultaneously: push, pop and findMin. Note that deleteMin is not part of the repertoire. Hint: Maintain two stacks – one to store items and the other to store minimums as they occur.

We can support the following operations in contant time by having two stacks that are synced. The first stack can call the main stack, and the second stack we can call the min stack. For each value we push into the main stack, we check to see if that value it lower than the current minimum value, and if so push that value onto the min stack. Else we push the current minimum onto the min stack. We can then find the minimum element by looking at the top element in the min stack. Min.peek().

1. Prove by induction the formula:

The Basis is true for F(0) and F(1).

Basis:

F(0) = 0

F(1) = 1

Induction Hypothesis:

Fn = Fn -1 + Fn -2

From F(2) = 1

F2 = F1 + F0

F2 = 1 + 0

1 = 1

So we assume it is true for all values such that 0 < i < n.

FN =

Fn = Fn-1 + Fn-2

X1 =

X2 =

Fn =

=

1. Solve the following recurrence, which in all cases have T(0) = T(1) = 1. A Big-Oh answer will suffice
   1. T(N) = T(N/2) + 1
   2. Solution is O(logn + 1)
2. Solve the following recurrence, which in all cases have T(0) = T(1) = 1. A Big-Oh answer will suffice
   1. T(N) = T(N/2) + log N
   2. T(n) = log(n) + log(n/2) + log(n/4) + … + log(1) + 1

= (log(n) – 0) + (log(n) – 1) + … + (log(n) – log(n)) + 1

= log(n) \* log(n) – log(n) \* (log(n) + 1)/2 + 1

= log(n)\*log(n)/2 – log(n)/2 + 1

= O(log2(n))