Data Structures and Algorithms

Fibonacci Heap Insertion

Printable Version

The *insert* routine simply adds a new node into the root list, updating a pointer to the minimum node in the root list.

```
function insert(h,x)    // h is the heap, x is the value
    {
    var n = new HeapNode(x);
    h.rootList = insertIntoList(n,h.rootList);
    h.size += 1;
    // keep track of the minimum, root list points to the new value
    if (h.size == 1 || h.min.value > x) h.min = h.rootList;
    }

// this is the standard insert into a circular, doubly-linked list routine
function insertIntoList(n,r) // r is the existing list
    {
    if (r == null)
        {
        n.next = n;
        n.prev = n;
        }
    else
        {
        n.next = r;
        n.prev = r.prev;
        r.prev.next = n;
        r.prev = n;
        }
    return n;
    }
```

Note that insertion happens in constant time and that the newly inserted node is at the "front" of the circular list.

Next: Extracting the minimum value from a *Fibonacci* heap