Chapter 19: Binomial Heaps

19.1-1) Suppose that x is a node in a binomial tree within a binomial heap, and assume that *siblings[x]*$\neq$ *NIL*. If x is not a root, how does *degree[siblings[x]]* compare to *degree[x]*? How about if x is a root?

Solution:
In a binomial tree, a parent node points to the leftmost child. The remaining children are in a linked list, where a child node has only one sibling, the next node that it points to (the node to the right). The child nodes are linked in a decreasing order based on degree. That is, "the children of the root are numbered from left to right by $k-1, k-2, ..., 0$, with parent degree k." Thus, degree[x] = degree[sinbling[x]] + 1 if sibling[x]$\neq$NIL.

If x is a root, degree[x]<degree[singling[x]] because binomial trees are linked in an increasing order within the binomial heap, which contains at most on binomial tree of degree k for a non-negative integer k.

19.1-2) If x is a nonroot node in a binomial tree within a binomial heap, how does *degree[x]* compare to *degree[p[x]]*?

Solution:

$$degree[x] = \left\{ \begin{array}{l} \text{If node is the left most child, } degree[x] = degree[p[x]] - 1 \\ \text{If node is the right most child, } dregree[x] = 0 \\ \text{Otherwise, } degree[p[x]] - 1 > degree[x] > 0 \end{array} \right\} degree[p[x]]-1 \geq degree[x] \geq 0$$

19.1-3) Suppose we label the nodes of binomial tree B_k in binary by a postorder walk, as in Figure 19.4. (a) Consider a node x labeled l at depth i, and let $j = k - i$. Show that x has j 1's in its binary representation.(b) How many binary k-strings are there that contain exactly j 1's? (c)Show that the degree of x is equal to the number of 1's to the right of the rightmost 0 in the binary represntation l.

Solution:

(a) The number of 1's is determined by the depth (or height) of the node.

Nodes at depth 0 (root) have k number of 1's	$j = k$	depth = 0
Nodes at depth 1 have $k-1$ number of 1's	$j = k-1$	depth = 1
Nodes at depth 2 have $k-2$ number of 1's	$j = k-2$	depth = 2
Nodes at depth 3 have $k-3$ number of 1's	$j = k-3$	depth = 3
$\vdots$	$\vdots$	$\vdots$
Nodes at depth i have $k-i$ number of 1's	$j = k-i$	depth = i
$\vdots$	$\vdots$	$\vdots$
Nodes at depth k have $k-k=0$ number of 1's	$j = k-k=0$	depth = k

In general, any node x with label l at dephth i has $j = k - i$ 1's in its binary representation.

(b) Each level has nodes labeled with j 1's. Since B_k has 2^k nodes, we need a minimum of $lg(2^k) = k$ bits. Each node at depth j has a binary label, l, that uses k bits. These nodes have j number of 1's. There are $\binom{k}{j}$ binary permutations. Thus, there are $\binom{k}{j}$ nodes at depth j and also $\binom{k}{j}$ binary k-strings that contain exactly j 1's.

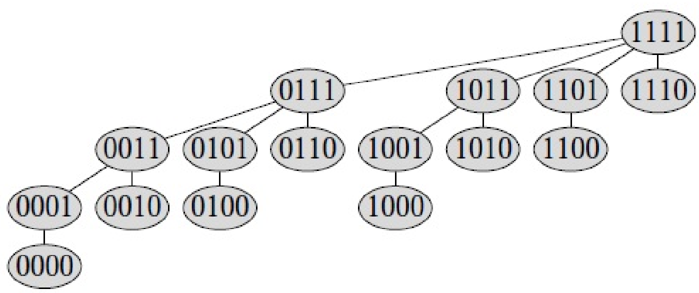

Figure 19.4 The binomial tree B_4 with nodes labeled in binary by a postorder walk.

(c) **Claim 1:** Let l be the binary label of the node x with its rightmost zero at position p. Let k be the number of bits as defined above. Then the children of x have bits p through k fixed. That is, if $l = \mathbf{b_k\ b_{k-1}}$ ... $\mathbf{b_{p+1}\ 0}$ 1...1, then a child of x has a binary label of the form $\mathbf{b_k\ b_{k-1}}$ ... $\mathbf{b_{p+1}\ 0}$ $b_{p-1}...b_1$.

If x has j ones in l, then the children of x must have $j-1$ ones in their binary representation. Claim 1 says that the leftmost bits of the descendents of x are fixed, but bits 1 through $p-1$ are not. Necessarily, the children of x must have a binary label where $p-2$ out of the $p-1$ right bits are 1's in order to ensure that the child nodes have $j-1$ ones (1's). There are $\binom{p-1}{p-2} = p-1$ permutations. Then node x must necessarily have $p-1$ children. Then node x must have degree $p-1$, which is equal to the number of 1's to the right of the rightmost 0 in the binary label l $\square$

Proof to Claim 1: (By Contradiction)
Let x_c be the child of x and let l_c be its label. Assume that l_c does not have the same $(k-p)$ left bits as its parent node x. Then the binary label, l_c, must contain $j-1$ ones (1's). Let l' be one of the $\binom{k}{j}$ labels containing j 1's with $l' \neq l$. Then l_c has the same label as l' with one of its '1' bits replaced by a 0 bit. Necessarily, l' must be a node with the same depth as l. Since l was used to write l_c, and l_c is the label of a child of x, then $l' < l$. More importantly, since l_c has one less '1' bit than l', then $l_c < l'$, which implies that $l_c < l' < l$. The nodes of a binary tree are labeled using postwalk order so the descendents of a parent node are bounded by any node to the left of the parent node (having the same depth as the parent) and the parent node. The node with the label l' is to the left of l. l_c is not bounded by l' and l. Thus, l_c cannot be a child of x. This is a contradiction because x_c is a child of x. Then our initial assumption that x_c does not have the same $(k-p)$ left bits as its parent node x must be false. Therefore, l_c must have the same $(k-p)$ left bits as its parent node x $\square$

References

[1] Cormen, Thomas. H., Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms, Second Edition*. MIT Press, Cambridge, MA, 2009.