CSCI 532 – Algorithm Design
Assignment 9

Name: Badarika Namineni **CWID:** 50233279

Question 1. For what values of t is the tree of Figure 18.1 a legal B-tree?

Solution:

We know that according to property 5 of B-tree, every node other than the root must have at least t−1 keys and at most 2t−1 keys. In Figure 18.1, the number of keys of each node except the root node is either 2 or 3. So to make it a legal B-tree, we need to guarantee that

$t - 1 \le 2$ and $2t - 1 \ge 3$, which yields $2 \le t \le 3$. So, t can be 2 or 3.

Question 2. As a function of the minimum degree t, what is the maximum number of keys that can be stored in a B-tree of height h?

Solution:

Let us consider we have 1 node at level 0, 2t nodes at level 1, $(2t)^2$ nodes at level 2 and so on. By this we know that each node contains 2t-1 keys, and at depth k the tree will have utmost $(2t)^k$ nodes. The total nodes is the sum of $(2t)^0$, $(2t)^1$, $(2t)^2$.. $(2t)^h$. MaxKeyNum (t,h) function returns the maximum number of keys in a B-tree of height h and the minimum degree t.

MaxKeyNum (t,h) = $(2t-1)[((2t)^0, (2t)^1, (2t)^2 .. (2t)^h]$

$= (2t-1) \sum_{i=0}^{h} (2t)^\wedge n$

$= (2t-1) * (2t)^{h+1} - 1/2t-1$

$= (2t)^{h+1} - 1$

Question 3. Describe the data structure that would result if each black node in a red-black tree were to absorb its red children, incorporating their children with its own.

Solution:

After absorbing each red node into its black parent, each black node may contain 1, 2 (1 red child), or 3 (2 red children) keys, and all leaves of the resulting tree have the same depth, according to property 5 of red-black tree (For each node, all paths from the node to descendant leaves contain the same number of black nodes). Therefore, a red-black tree will become a B tree with minimum degree t = 2, i.e., a 2-3-4 tree.

Question 4. Show the d and PI values that result from running breadth-first search on the directed graph of Figure 22.2(a), using vertex 3 as the source.

Solution:

Vertex	1	2	3	4	5	6
D	∞	3	0	2	1	1
PI	NIL	4	NIL	5	3	3

Question 5. Show that using a single bit to store each vertex color suffices by arguing that the BFS procedure would produce the same result if lines 5 and 14 were removed.

Solution:

The BFS procedure cares only whether a vertex is white or not. A vertex v must become non-white at the same time that v.d is assigned a finite value so that we do not attempt to assign to v.d again, and so we need to change vertex colors in lines 5 and 14. Once we have changed a vertex's color to non-white, we do not need to change it again.

Question 6. As a function of the minimum degree t, what is the maximum number of keys that can be stored in a B-tree of height h?

Solution:

Let us consider we have 1 node at level 0, 2t nodes at level 1, $(2t)^2$ nodes at level 2 and so on. By this we know that each node contains 2t-1 keys, and at depth k the tree will have utmost $(2t)^k$ nodes. The total nodes is the sum of $(2t)^0$, $(2t)^1$, $(2t)^2$.. $(2t)^h$. MaxKeyNum (t,h) function returns the maximum number of keys in a B-tree of height h and the minimum degree t.

MaxKeyNum (t,h) = (2t-1)[($(2t)^0$, $(2t)^1$, $(2t)^2$.. $(2t)^h$]

= (2t-1) $\sum_{i=0}^{h}(2t)^\wedge n$

= (2t-1) * $(2t)^{h+1}$-1/2t-1

=$(2t)^{h+1}$-1