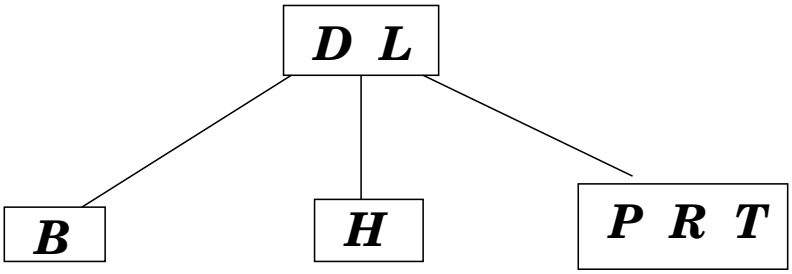
Data Structures and Algorithms (CS 130A) **Prof. Suri**

Homework Assignment 4

Handed Out: March 1 **Due:** *March 8*

1. (10 pts) Consider the following B-Tree, where the branching factor is $t = 2$. Suppose we insert the keys Y, F, X, and Z (in that order) into this tree. Show the new B-Tree after each insertion.



2. (20 pts) Show the result of the following sequence of instructions:
   union(1,2), union(3,4), union(3,5), union(1,7),
   union(3,6), union(8,9), union(1,8), union(3,10),
   union(3,11), union(3,12), union(3,13), union(14,15),
   union(16,0), union(14,16), union(1,3), union(1,14),
   when the unions are

   - Performed arbitrarily to achieve worst case tree heights,
   - Performed by height (always making the shallower tree the child),
   - Performed by size (always making the smaller tree the child).

   Finally, for each of these (final) trees, perform a find with path compression on the deepest node, and show the resulting tree.

3. (10 pts) Suppose we start with n singleton nodes, and perform an arbitrary sequence of unions using the **union by height** rule—that is, always make the shallower tree the child. Then, give a proof (reasoning) that the worst-case depth of any tree is $O(\log n)$.

4. (10 pts) Give a proof for the following assertion: *if all of the unions precede all the finds, then the disjoint set algorithm with path compression requires $O(n)$ time,* **even if unions are done arbitrarily.**