Data Structure Final Exam.
Jan. 15, 2009

1. Union-Find operations implemented by using forest.
   (a) What are the collapsing rule for Find and weighting rule for Union?
   (b) Show that the height of the tree is bounded above by $O(\log n)$ if only the weighting rule for Union is applied.
   (c) Show that the bound $O(\log n)$ is tight (there is a case that the resulted tree has height $\log n$ when weighting rule is applied).
   (d) If both weighting rule and collapsing rule are applied, what is the time required for a sequence of $n-1$ Unions and m Finds.

2. Optimal Binary Search Tree: You have n records a_i, $i = 1, \ldots, n$ with different access probabilities stored in a binary search tree. $c_{i,j}$ is the cost for the optimal binary search tree $T_{i,j}$.
   (a) Derive the recursion $c_{i,j} = w_{i,j} + \min_{i<k\leq j}(c_{i,k-1} + c_{k,j})$.
   (b) If there are 4 records a_i, $i = 1, \ldots, 4$. Let $(p_1, p_2, p_3, p_4) = (3, 3, 1, 1)$ and $(q_0, q_1, q_2, q_3, q_4)$ $= (2, 3, 1, 1, 1)$. Compute $w_{0,4}$, $c_{0,4}$, and $r_{0,4}$. (Figure 1)
   (c) Reconstruct the optimal binary search tree $T_{0,4}$ from the computed $r_{i,j}$.
   (d) The optimal binary search tree $T_{i,j}$ consists of 2 smaller optimal binary search trees. Why we cannot solve the problem by using the divide and conquer technique.

3. Given a graph, G, as shown in Figure 2,
   (a) Please give me the adjancency list representation of the graph.
   (b) Depth first search the graph using your adjancy list representation starting vertex 0. Draw the depth first searh tree and mark down the *dfn* for each node.
   (c) Compute the *low* for each vertex and determine the articulation point(s).

4. Given a graph G, an AOV network, as shown in Figure 3, topological sort the vertex.

5. Why does any comparison sorting algorithm need at least $\Omega(n \log n)$ operations to sort n records.

6. Show that the least significant digit fist radix sort algorithm correctly sorts n records in Linear time (Note that there are two parts, first part is "linear time", 2nd part is "correctly sort"). Why does this time required beat $\Omega(n \log n)$ lower bound established previously.

7. Describe the merge sort algorithm. Show that merge sort n records can be done in $O(n \log n)$ time.

8. Given a Leftist tree as shown in Figure 4, delete minimum from the Leftist tree. Argue that the operation can be done in $O(\log n)$ time.

9. Given the binomial heap shown in Figure 5, delete the minimum from the binomial heap. Argue that the operation can be dont in $O(\log n)$ time.

10. Given the AVL-tree shown in Figure 6, insert 47 then insert 80 into the AVL-tree.

11. Can you delete 50 from the resulted tree from previous problem?

	0	1	2	3	4
0	$w_{00} = 2$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = 3$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = 1$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1$ $c_{44} = 0$ $r_{44} = 0$
1	$w_{01} = 8$ $c_{01} = 8$ $r_{01} = 1$	$w_{12} = 7$ $c_{12} = 7$ $r_{12} = 2$	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = 3$	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = 4$	
2	$w_{02} = 12$ $c_{02} = 19$ $r_{02} = 1$	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = 2$	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = 3$		
3	$w_{03} = 14$ $c_{03} = 25$ $r_{03} = 2$	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = 2$			
4	w_{04} : c_{04} : r_{04} :				

Figure 1.

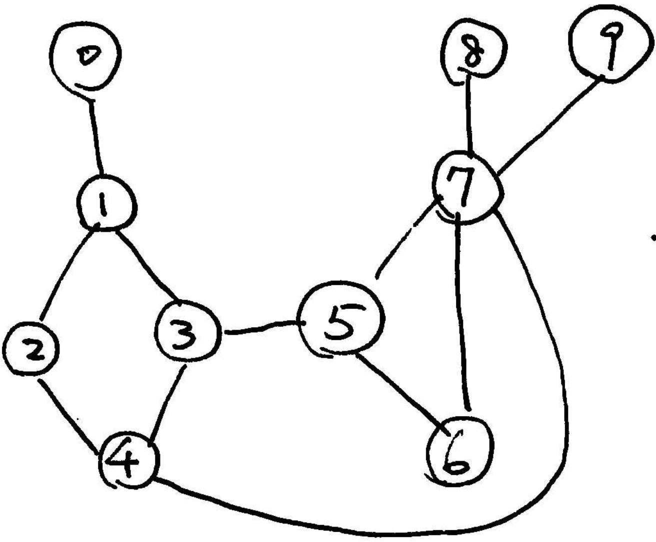

Figure 2.

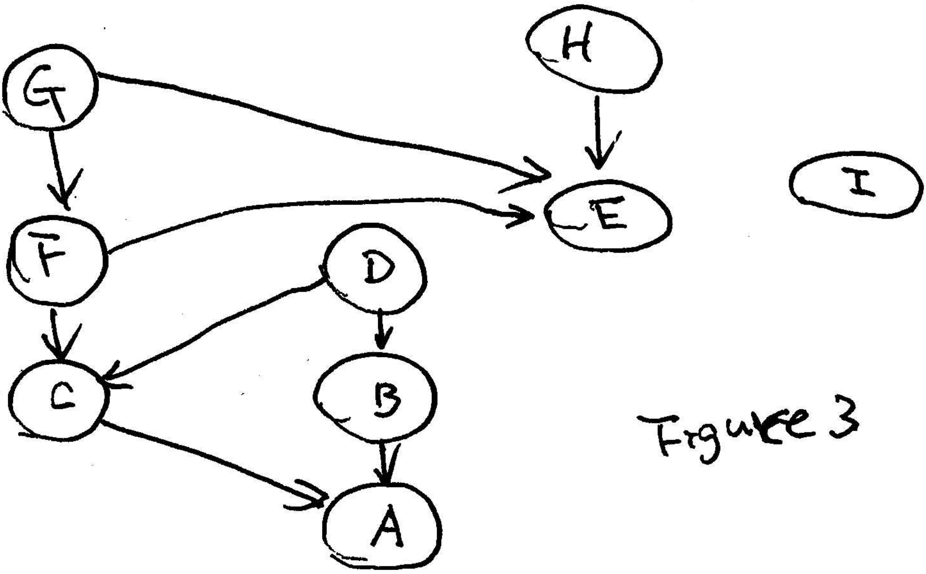

Figure 3

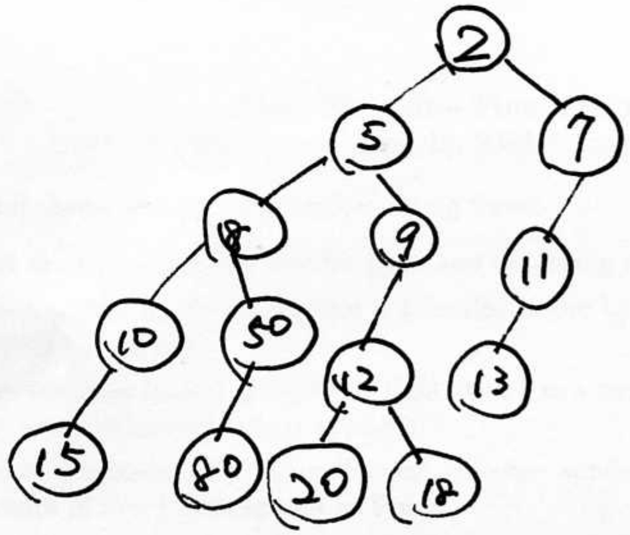

Figure 4.

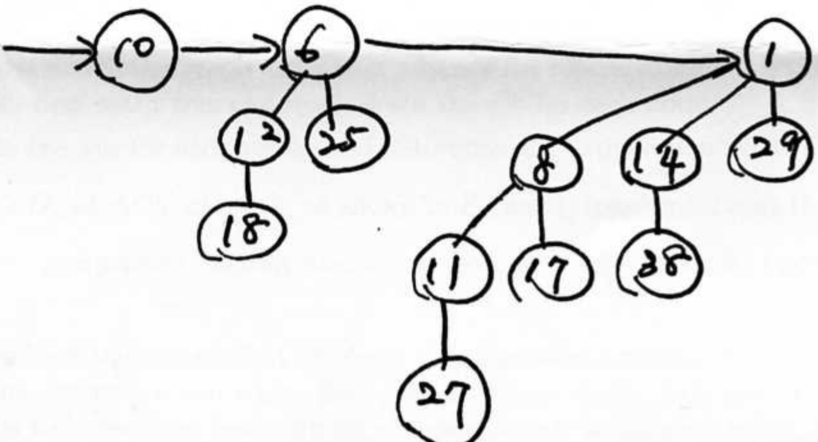

Figure 5

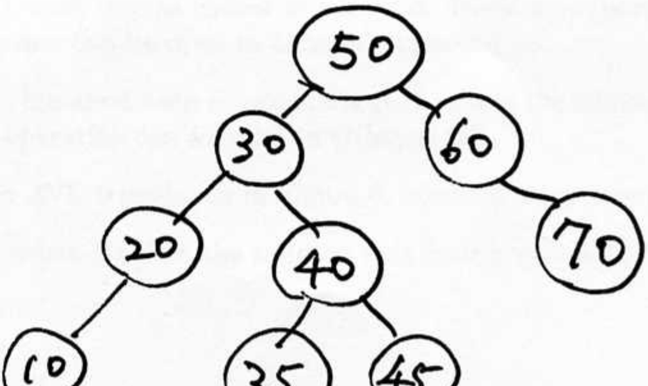

Figure 6