Analysis of Algorithms

Final Exam : 16 June 2011

1. Compute a longest common subsequence of $\langle a, b, b, a, b, a, b, a \rangle$ and $\langle b, a, b, a, a, b, a, a, b \rangle$.

2. Describe Huffman's algorithm (with its input and output). Explain its time complexity.

3. Show how the procedure STRONGLY-CONNECTED-COMPONENTS works on the following graph G. Show the discover/finish times computed in DFS(G). Assume that DFS considers vertices in alphabetical order and that the adjacency lists are in alphabetical order.

4. The *edge connectivity* between vertices u and v in an undirected graph G is the minimum number of edges that must be removed to disconnect u and v. Show how the edge connectivity between two vertices can be solved by running a maximum flow algorithm.

5. Show that if an algorithm makes at most a constant number of calls to polynomial-time subroutines and performs an additional amount work that also takes polynomial time, then it runs in polynomial time. Also show that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

6. Choose another team's presentation. Describe its problem, the algorithm outline, and time complexity.