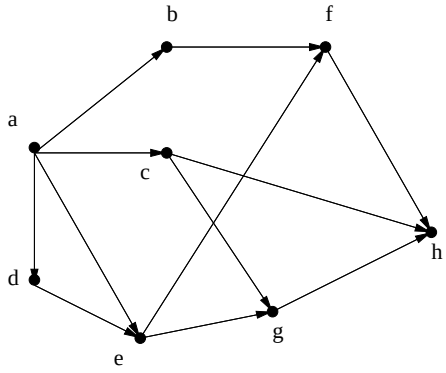
Mid-semester exam
Design and Analysis of Algorithms (January-April 2012)

23rd February, 2012

For each algorithm, write a correctness proof and analyze the time complexity.

1. (3 marks) What is the running time of heapsort when the input is completely sorted? What about reverse sorted input?

2. (8 marks) In the following graph, perform a depth-first search and determine the discovery and finish time of each vertex. Use this information for topological sorting of the graph. Also list forward edges, back edges, and cross edges.



3. (8 marks) It takes $O(n+m)$ time to merge two sorted lists of length n and m respectively. Consider the problem of pairwise merging k sorted lists of lengths $n_1, \ldots, n_k$. Thus, we pick two lists from the given k lists and merge them. This results in $k-1$ lists. We continue this process to finally get a single list.

   (a) Does the time complexity depend on the order in which the lists are merged?

   (b) If the answer to the above question is *yes*, describe an algorithm to decide the order in which the lists should be merged so as to minimize the time taken.

4. (8 marks) Given a sequence of n real numbers $a_1, a_2, \ldots, a_n$, describe an algorithm to determine a contiguous subsequence $a_i, \ldots, a_j$ that has the maximum sum.

5. (5 marks) Given a directed acyclic graph G and two vertices s and t, give a linear time algorithm to determine the number of paths from s to t in G. (Note that you need not list the paths.)

6. (8 marks) Prove or disprove the following:

   (a) In a breadth-first search of a directed graph, there are no forward edges.

   (b) In a depth-first search of an undirected graph, there are no cross edges.