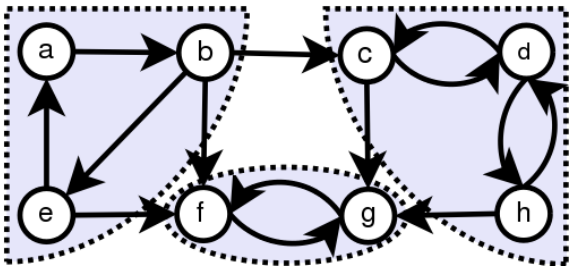
CS180 Homework 3

Due: 4:00pm, 04/20/2016

1. A Strongly Connected Component (SCC) S in a directed graph $G = (V, E)$ is a maximal set of vertices $S \subseteq V$ such that for every pair of vertices u and v in S, they are mutually reachable from each other. A directed graph may contain multiple SCCs. If we can decompose a directed graph into its SCCs, we can build a SCC graph: create a node for each SCC and assign a directed edge $S_i \to S_j$ if there is a node p in the SCC S_i has a directed edge to a node q in SCC S_j in the original graph.



A directed graph with eight nodes and corresponding three strongly connected components: $S_1 = \{a, b, e\}, S_2 = \{f, g\}, S_3 = \{c, d, h\}$. So the SCC graph has three nodes S_1, S_2 and S_3 and three edges $S_1 \to S_2, S_1 \to S_3, S_3 \to S_2$.

   (a) Prove that the SCC graph is a DAG (Directed Acyclic Graph).

   (b) Given a directed graph $G = (V, E)$ and all nodes can be reached from node v. Write an algorithm to build the SCC graph. Prove the correctness and the time complexity. Your algorithm should be in linear time.
   (Hint:
      i. Do DFS on node v and assign post-order numbers to all nodes.
      ii. Do another DFS against the edges on the node with the largest post-order number. The nodes the DFS can reach belongs to a SCC.
      iii. Remove the nodes in the SCC and repeat the second step.)

2. The longest path in a Directed Acyclic Graph (DAG) is a path that has the maximum length.

   (a) Given a DAG with unweighted edges, the length of a path is the number of edges in the path. Write an algorithm that finds the longest path in such a DAG.

   (b) Given a DAG with weighted edges and the weights of edges are real numbers, the length of a path is the sum of weights of the edges in the path. Write an algorithm that finds the longest path in such a DAG.

   (c) Now consider a scheduling problem for a project which consists of n jobs. Each job i has an execution time t_i, and a precondition that specifies jobs i can not be started before some jobs are finished. The precondition relations between jobs are acyclic. Multiple jobs can be executed at the same time if the preconditions for those jobs are satisfied. Give an algorithm to find a schedule that minimized the total time to finish all jobs. (Hints: we can think each job as a node in a DAG. An edge $u \to v$ represents job v can not be started before finishing job v.)

3. We have a number of n files, and file i has a length ℓ_i and probability p_i that whether it is accessed, and we want to write these files on a tape in some order. We assume that every time we need to access a file, we have to start from the beginning of tape until we reach the start of the requested file, so the time to access a file is proportional to the total length of all the files that are saved before it. Design an algorithm that finds the optimal order of files on the tape. The optimal order is an order that minimizes the average access time. The average access time is $\sum_{i=1}^{n} d_i p_i$ where d_i is distance of the beginning of file i from the beginning of the tape (the total length of all the files that is saved before file i).

4. In class we have seen the problem of scheduling jobs where each job i has a length of execution p_i, and a deadline d_i. Somebody built an algorithm SC which gives such optimal scheduling. She claims that the time complexity of her algorithm is $o(n \log n)$ (little o stands for a function that is asymptotically below $n \log n$). Show how you can use SC to design a sorting algorithm with $o(n \log n)$ time complexity.

★ Homework assignments are **STRICTLY** due on the exact time indicated. Please submit a hard copy of your homework solution with your **Name**, **Bruin ID**, **Discussion Number**, clearly indicated on the first page. If your homework consists of multiple pages, please **staple** them together. Email attachments or other electronic delivery methods are not acceptable.

★ We recommend to use LaTeX, LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receiver fewer points.

★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.