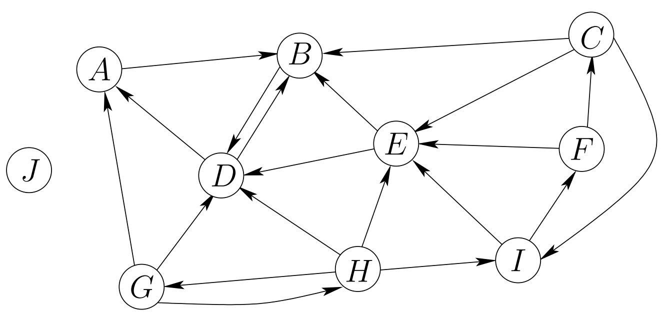
CSE100 Algorithm Design and Analysis — Homework set #4
Fall semester 2009 — Miguel Á. Carreira-Perpiñán

Total possible marks: 100. Homeworks must be solved individually. This set covers chapters 22–24 of the textbook *Introduction to Algorithms*, 3rd. ed., by Cormen et al. References to pages and figures correspond to the textbook.

Exercise 1: breadth-first search (20 points).

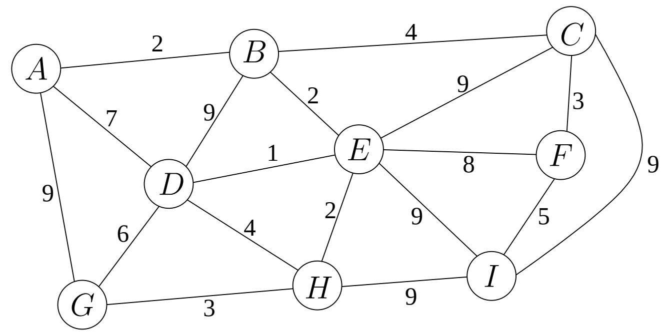
1. (8 points) The breadth-first search algorithm of p. 595 explores only vertices reachable from a given starting vertex. Modify it as in the DFS algorithm of p. 604 to explore all vertices.
2. (6 points) Modify this BFS pseudocode into a BFS-CONNECTED-COMPONENTS(G) algorithm that finds all the connected components of an undirected graph G (represented by adjacency lists). *Hint*: see exercise 22.3-12 for undirected connected components with depth-first search.
3. (6 points) Give the running time of BFS-CONNECTED-COMPONENTS(G).

Exercise 2: depth-first search, topological sort and strongly connected components (35 points). Consider the following directed graph $G = (V, E)$:



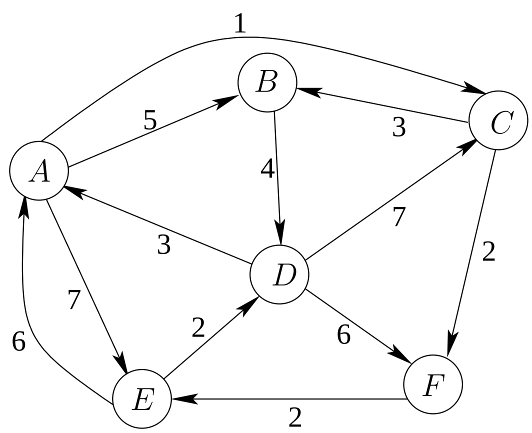
1. (3 points) Draw the adjacency-list representation of G as in fig. 22.2, with each list sorted in increasing alphabetical order.
2. (2 points) Give the adjacency matrix of G.
3. (5 points) Draw the graph, the adjacency-list representation (with each list sorted in increasing alphabetical order), and the adjacency matrix for the transpose graph G^T.
4. (8 points) Do depth-first search in G, considering vertices in increasing alphabetical order. Show the final result, with vertices labeled with their starting and finishing times, and edges labeled with their type (T/B/F/C) as in fig. 22.5(a).
5. (8 points) Based on your results, proceed to run the algorithm in p. 617 to find the strongly connected components of G (show the result of the DFS in G^T, with vertices labeled with their starting and finishing times).
6. (4 points) Draw the component graph G^{SCC} of G. Is it a dag/tree/forest/none of these?
7. (5 points) Find a topological sort of G^{SCC} using the algorithm in p. 613 (label each vertex with its DFS finishing time).

Exercise 3: minimum spanning trees (20 points). Consider the following weighted undirected graph:



1. (10 points) Show the intermediate stages of Kruskal's algorithm as in fig. 23.4.
2. (10 points) Show the intermediate stages of Prim's algorithm as in fig. 23.5; use A as initial vertex.

Exercise 4: shortest paths (25 points). Consider the following weighted directed graph:



1. (13 points) Show the intermediate stages of Dijkstra's algorithm as in fig. 24.6 to find the single-source shortest paths starting from vertex A.
2. (12 points) Consider the same graph but ignoring the weights. Use breadth-first search starting from vertex A to label each vertex (as in fig. 22.3 in the book) with its shortest distance (in number of edges) from A, and mark the edges in the BFS tree.

Bonus exercise 1: minimum spanning trees (10 points). (Exercise 23.1-1 in the textbook.) Let (u, v) be a minimum-weight edge in a connected graph G. Show that (u, v) belongs to some minimum spanning tree of G. *Hint*: consider theorem 23.1 and corollary 23.2.

Bonus exercise 2: minimum spanning trees (15 points). (Exercise 23.2-2 in the textbook.) Suppose that we represent the graph $G = (V, E)$ as an adjacency matrix. Give a simple implementation of Prim's algorithm for this case that runs in $\mathcal{O}(V^2)$ time.