THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI.220.C.S1 – Algorithms and Data Structures

Assignment 4 – Graph algorithms

Out: Monday, 9$^{\text{th}}$ of May, 2016

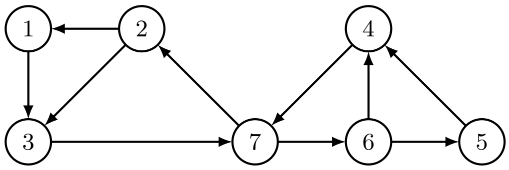
Due: Monday, 30$^{\text{th}}$ of May, 2016

Assignment 4 (Lectures 18 – 30) is worth **70 marks** representing **7%** of your total course grade.

Objectives. Learning unweighted and weighted (di)graphs and algorithms to traverse these data structures, find their shortest paths and minimum spanning trees, place their nodes in topological order, and extend matchings in a bipartite graph.

Requirements. You should give clear and detailed answers to the following questions.

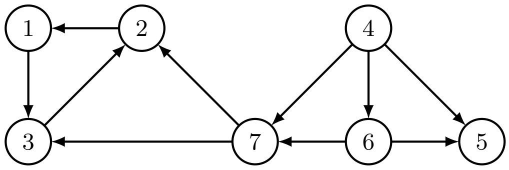
1. (*10 marks*) Suppose a digraph $G = (V, E)$ below with the sets $V = \{1, 2, 3, 4, 5, 6, 7\}$ of nodes and $E = \{(1, 3), (2, 1), (2, 3), (3, 7), (4, 7), (5, 4), (6, 4), (6, 5), (7, 2), (7, 6)\}$ of arcs:



   (a) Represent G with adjacency lists and an adjacency matrix for numerically ordered nodes $1, 2, \ldots, 7$ (see Example 4.23 of Textbook and Lectures 18–20).

   (b) Determine the order, size, diameter, radius, girth, and directed girth of G with references to related definitions in Textbook and Lectures 18–20.

2. (*10 marks*) Suppose that the depth-first search algorithm `DFS` from Textbook (Section 5.3, Fig. 5.7) is run on another digraph $G = (V, E)$ below with the sets $V = \{1, 2, 3, 4, 5, 6, 7\}$ of nodes and $E = \{(1, 3), (2, 1), (3, 2), (4, 5), (4, 6), (4, 7), (6, 5), (6, 7), (7, 2), (7, 3)\}$ of arcs:



   The DFS starts at the node 1 and selects either a next adjacent node, or a next tree root in the search forest in ascending numerical order.

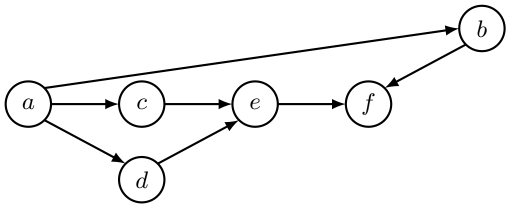
   (a) Compute and tabulate (like in Exercise 5.3.5 of Textbook) timestamps $seen[v]$ and $done[v]$ for all the nodes $v \in V = \{1, 2, 3, 4, 5, 6, 7\}$.

   (b) Classify each arc $e \in E$ as a tree arc (ta) in the DFS forest, or a forward arc (fa), or a cross arc (ca), or a back arc (ba); justify your classification, and tabulate the arcs and classes:

$e \in E$	$(1,3)$	$(2,1)$	$(3,2)$	$\ldots$	$(7,3)$
Class C_e	$C_{1,3}$	$C_{2,1}$	$C_{3,2}$	$\ldots$	$C_{7,3}$

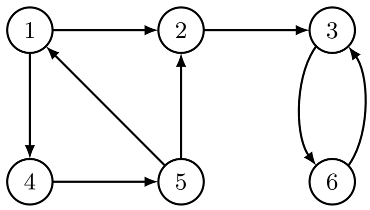
   where $C_e \in \{\text{ta}, \text{fa}, \text{ca}, \text{ba}\}$.

3. (*10 marks*) Suppose a directed acyclic graph $G = (V, E)$ below with the sets $V = \{a, b, c, d, e, f\}$ of nodes and $E = \{(a, b), (a, c), (a, d), (b, f), (c, e), (d, e), (e, f)\}$ of arcs:



   Assuming the DFS visits adjacent nodes in alphabetical order, find a topological order of the nodes $v \in V$ by running the DFS on this DAG G from the source (zero in-degree) node and describe your solution in detail, including the DFS timestamps and their relation to the goal topological ordering.

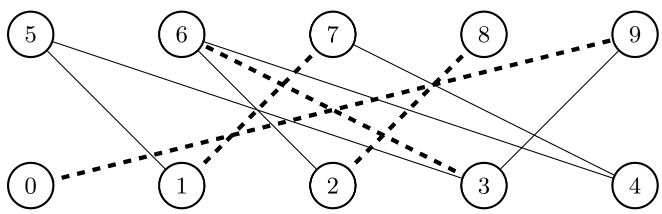
4. (*10 marks*) Consider a digraph $G = (V, E)$ below with the sets $V = \{1, 2, 3, 4, 5, 6\}$ of nodes and $E = \{(1, 2), (1, 4), (2, 3), (3, 6), (4, 5), (5, 1), (5, 2), (6, 3)\}$ of edges:



and find its strong components (i.e., strongly connected components) by using the algorithm from Textbook (Section 5.7) and Lectures 21–23, which runs in linear time when the digraph G is represented with adjacency lists.

Describe your solution in detail, including the corresponding reverse digraph G_{r}; timestamps of the first DFS of G for topological ordering of its nodes; timestamps of the second DFS on G_{r} that selects the roots of each new search tree with due account of their topological order found, and the resulting strong components.

5. (*10 marks*) A bipartite graph $G = (V, E)$ has the sets $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ of vertices and $E = \{(0, 9), (1, 5), (1, 7), (2, 6), (2, 8), (3, 5), (3, 6), (3, 9), (4, 6), (4, 7)\}$ of edges:

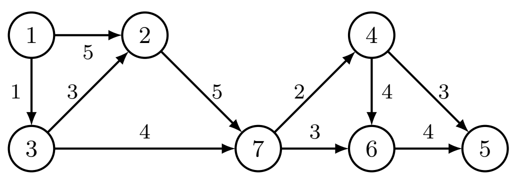


Given a maximal matching $M^\circ = \{(0, 9),\ (1, 7),\ (2, 8),\ (3, 6)\}$ of the four pairs of vertices (dashed edges), find two maximum matchings of all five pairs of vertices as follows:

(a) Select two possible augmenting paths P_1 and P_2 for M° and justify your selection.

(b) Derive each goal maximum matching M_i from M° and the path P_i; $i = 1, 2$.

6. (*10 marks*) A weighted digraph $\big(G = (V, E), c\big)$ below has the sets $V = \{1, 2, 3, 4, 5, 6, 7\}$ of nodes and $E = \{(1, 2), (1, 3), (2, 7), (3, 2), (3, 7), (4, 5), (4, 6), (6, 5), (7, 4), (7, 6)\}$ of arcs:

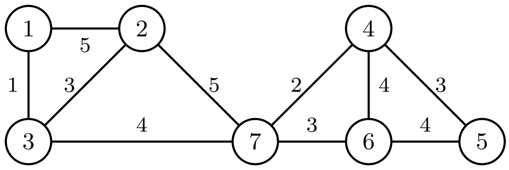


and the following weights, or costs $c(e)$; $e \in E$, of the arcs:

e	$(1,2)$	$(1,3)$	$(2,7)$	$(3,2)$	$(3,7)$	$(4,5)$	$(4,6)$	$(6,5)$	$(7,4)$	$(7,6)$
$c(e)$	5	1	5	3	4	3	4	4	2	3

Let algorithm `Dijkstra` from Textbook (Section 6.5, Fig. 6.2) be run on this weighted graph to find the minimum-weight paths from the source node 1 to each other node v. Illustrate successive steps of this algorithm by tabulating at each step the set S of the already visited nodes, which grows from $S = \{1\}$ to $S = V$, and its corresponding distance vector *dist* in the same manner as for the source node 0 in Table 6.1 of Textbook.

7. (*10 marks*) A weighted graph $\big(G = (V, E), c\big)$ below has the sets $V = \{1, 2, 3, 4, 5, 6, 7\}$ of vertices and $E = \{(1, 2), (1, 3), (2, 3), (2, 7), (3, 7), (4, 5), (4, 6), (4, 7), (5, 6), (6, 7)\}$ of edges:



and the following weights, or costs $c(e)$; $e \in E$, of the edges:

e	$(1,2)$	$(1,3)$	$(2,3)$	$(2,7)$	$(3,7)$	$(4,5)$	$(4,6)$	$(4,7)$	$(5,6)$	$(6,7)$
$c(e)$	5	1	3	5	4	3	4	2	4	3

Let algorithm `Kruskal` from Textbook (Section 6.5, Fig. 6.8) be run on this graph to build its minimum spanning tree. After initialisation, a disjoint-sets ADT A consists of seven single-vertex sets: $A = \big\{\{\underline{1}\};\ \{\underline{2}\};\ \{\underline{3}\};\ \{\underline{4}\};\ \{\underline{5}\};\ \{\underline{6}\};\ \{\underline{7}\}\big\}$ and all edges, $e \in E$, are sorted in ascending order of their costs, $1, 2, 3, 3, 3, 4, 4, 4, 5, 5$, i.e.,

$$(1,3);\ (4,7);\ (2,3);\ (4,5);\ (6,7);\ (3,7);\ (4,6);\ (5,6);\ (1,2);\ (2,7)$$

Perform all steps of the main `Kruskal`'s **for**-loop for the edges in increasing cost:

(a) At each step, list the current edge $e = (u, v)$ under consideration; indicate the disjoint sets, $S_u = A.\texttt{set}(u)$ and $S_v = A.\texttt{set}(v)$ that the nodes u and v are in, respectively, and explain whether and why this edge will be added or not to the goal MST.

(b) If the edge is added, list all sets of vertices in A after their merging $A.\texttt{union}(S_u, S_v)$.

(c) Finally, list all the edges forming the MST and give their total cost.

Submission: Your report should answer in detail to Questions 1–7. The report should be submitted as a single Adobe PDF file `CS220assign4.pdf` (only this file will be marked, so check that it can be read by PCs in the departmental Computer Labs). **Scanned handwritten documents are strictly forbidden** (even as images in a pdf file) and will not be accepted for marking. However, scanned images of hand-drawn (di)graphs can be used to clarify your report, despite it is more professional to prepare illustrations with any graphical editor.

Submit your file electronically to Assignment Drop Box (`https://adb.auckland.ac.nz`) (**not to Canvas!**) before Monday, 30th of May 2016, 09:00 pm (ADB time). If submitted after this due date, the penalty of 10% will be before 31st of May 2016, 09:00 p.m.; then the penalty of 50% will be before 1st of June 2016, 09:00 p.m., and no submission afterwards.

Marking scheme	% of marks
Clear structure of your report and detailed explanations	up to 20
Correctness of your final answers	up to 20
Correctness of your intermediate steps in deriving these answers	up to 20
Detailing all your steps with references, if necessary, to the textbook	up to 40
	Total: up to 100