York University
LE/EECS 3101 A, Fall 2023
Assignment 5

Due Date: December 11th, at 11:59pm

The August noon in us works to stave off the November chills ...

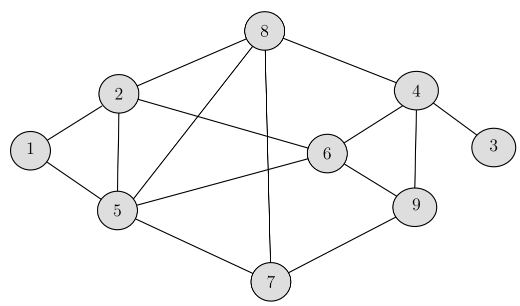
Ray Bradbury

All problems are written problems; submit your solutions electronically **only via Crowdmark**. You are welcome to discuss the general idea of the problems with other students. However, you must write your answers individually and mention your peers (with whom you discussed the problems) in your solution.

Problem 1 Graph Traversals [3 + 3 + 3 = 9 marks]

The following questions concern traversing a graph G with n vertices. We assume vertices are labelled $1, 2, \ldots, n$, and G is represented with a sorted adjacency list.

a) Apply the Depth-First-Search (DFS) algorithm in the following graph. It suffices to show the ordering in which the vertices are visited. Also, show the state of the stack (the elements inside the stack) when node 9 is visited.
Assume the traversal starts at vertex 1.



Answer: The DFS order is $1, 2, 5, 6, 4, 3, 8, 7, 9$. The stack contains $1, 2, 5, 6, 4, 8, 7$ at the time of visiting 9 (both "$1, 2, 5, 6, 4, 8, 7$" and "$1, 2, 5, 6, 4, 8, 7, 9$" are acceptable for the state of the stack.).

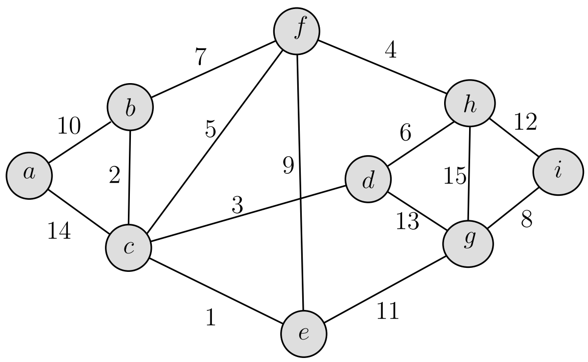
b) Apply the Breadth-First-Search (BFS) algorithm in the same graph of part (a). It suffices to show the ordering in which the vertices are visited. Also, show the state of the queue (the elements inside the queue in the order they will be dequeued) when

node 7 is enqueued.
Assume the traversal starts at vertex 1.

Answer: The BFS order is $1, 2, 5, 6, 8, 7, 4, 9, 3$. Node 7 is enqueued after 5 is dequeued, at which point the queue contains $6, 8$. (Both "$6, 8$" and "$6, 8, 7$" are acceptable.)

Problem 2 Minimum Spanning Trees Algorithms [3 + 3 = 6 marks]

Consider the following graph:



(a) write down the first 6 edges that are selected by the Kruskal's algrotithm. You just need to provide a list of edges (indicate each edge with its endpoints, e.g., (a,b)).

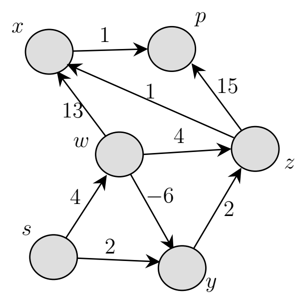
(b) write down the first 6 edges that are selected by the Prim's algrotithm. You just need to provide a list of edges (indicate each edge with its endpoints, e.g., (a,b)).

Answer: Kruskal: $(c, e), (b, c), (c, d), (f, h), (c, f), (g, i)$
Prim: $(c, e), (b, c), (c, d), (c, f), (f, h), (a, b)$

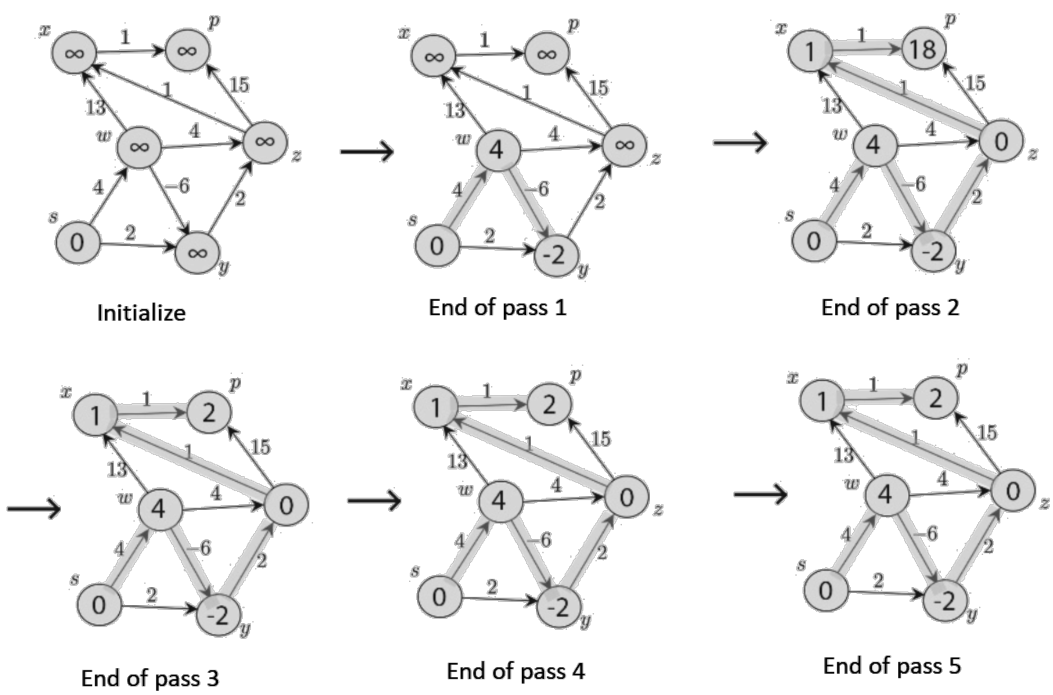
Problem 3 Bellman-Ford Algorithm [5 marks]

Follow the steps of the Bellman-Ford algorithm on the following directed graph, using vertex s as the source. In each pass, relax edges in the following order, and show the d and π values after each pass (after each iteration). For the π values, it suffices to shade or color the edges from predecessors to nodes. The algorithm makes five passes and you need to show the values at the end of each pass.

$$(z,p).(w,x),(x,p),(w,z),(y,z),(z,x)(s,w),(s,y),(w,y)$$

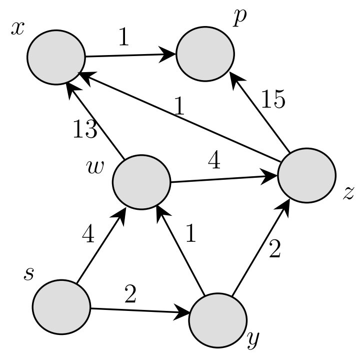


Answer: See below:

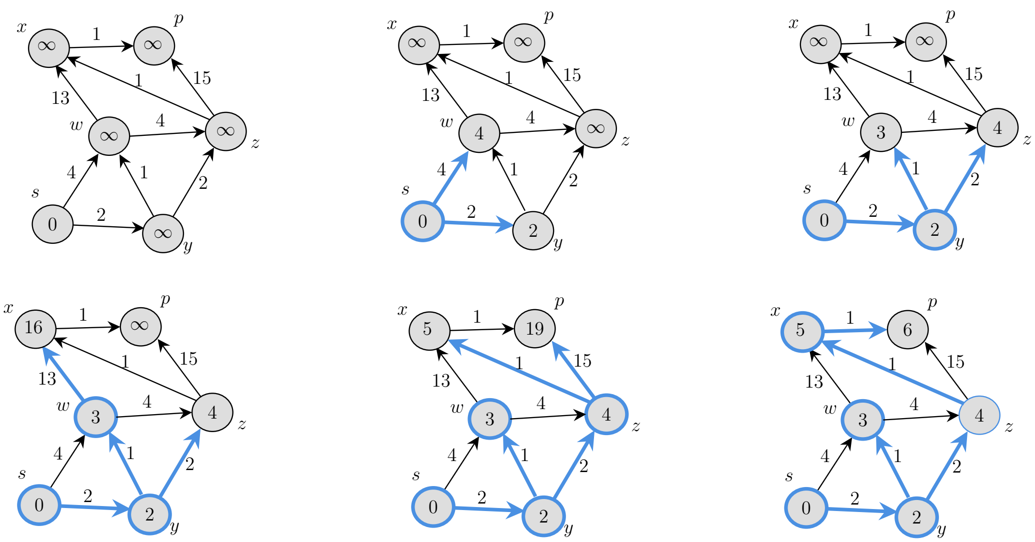


Problem 4 Dijkstra Algorithm [5 marks]

Follow the steps of the Dijkstra's algorithm on the following directed graph, using vertex s as the source. Show the d and π values after each pass (after processing each vertex). For the π values, it suffices to shade or color the edges from predecessors to nodes.

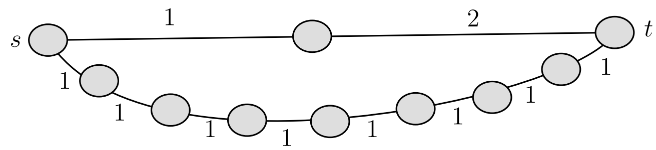


Answer: See below:



Problem 5 More Shortest Paths [4 marks]

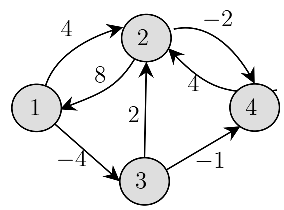
Consider a variant of the shortest path problem in which we are given a weighted graph with positive integer weights and a source vertex s. We want to the know the length of the *nicest path* from s to any other vertex. Here, the nicest path between s and t is the path in which the product of the edge lengths is minimized. For example, in the figure below, the nicest path between s and t is of product 1. Describe an algorithm to find the nicest path. You need to explain an algorithm in a few English words and justify why your algorithm works.



Answer: Just apply Dijkstra's algorithm. As before, maintain a value for each node which is initially 0 for s and ∞ for any other vertex. Initiate a priority queue, based on the stored values, that contains all vertices. Repeatedly take the vertex with minimum value, say p from the priority queue and relax over its outgoing edges as follows. Let (p, q) be an outgoing edge of p. In the relax procedure, we update the value of q to be $\min\{val(q), val(p) * w(p, q)\}$. When we dequeue a vertex p at iteration i, we know that the nicest path from s to p using i hubs is of length $val(p)$. Any path of length more than i must have larger product because it uses more than i hubs and extra hubs do not decrease the product (given that weights are positive integers).

Problem 6 All-pair Shortest Path [6 marks]

We have followed the steps of the Floyd-Warshall algorithm for finding the weight of all-pair shortest paths in the following graph. Most values in matrices D_i are provided for $i \in \{0, 4\}$. A few indices, however, are missing, and you should calculate them. It suffices to write down the missing values in the tables.



	1	2	3	4
1	0			
2	8	0	∞	-2
3	∞	2	0	-1
4		4	∞	0

D^0

	1	2	3	4
1	0			
2	8	0	4	-2
3	∞	2	0	-1
4		4	∞	0

D^1

	1	2	3	4
1	0			
2	8	0	4	-2
3	10	2	0	-1
4		4	8	0

D^2

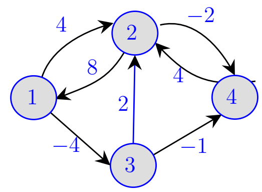
	1	2	3	4
1	0			
2	8	0	4	-2
3	10	2	0	-1
4		4	8	0

D^3

	1	2	3	4
1	0			
2	8	0	4	-2
3	10	2	0	-1
4		4	8	0

D^4

Answer: See the following:



	1	2	3	4
1	0	**4**	**-4**	∞
2	8	0	∞	-2
3	∞	2	0	-1
4	∞	4	∞	0

D^0

	1	2	3	4
1	0	**4**	**-4**	∞
2	8	0	4	-2
3	∞	2	0	-1
4	∞	4	∞	0

D^1

	1	2	3	4
1	0	**4**	**-4**	2
2	8	0	4	-2
3	10	2	0	-1
4	12	4	8	0

D^2

	1	2	3	4
1	0	**-2**	**-4**	**-5**
2	8	0	4	-2
3	10	2	0	-1
4	12	4	8	0

D^3

	1	2	3	4
1	0	**-2**	**-4**	**-5**
2	8	0	4	-2
3	10	2	0	-1
4	12	4	8	0

D^4