Algorithms: Final Assignment

Spring 2021

Due: 20 June 2021

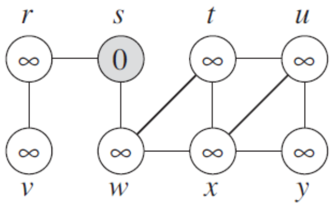
Submit a report containing outputs of the following problems and the codes (for P1, P2, P4, P6, using C or Python) you have written (screenshots okay). For P3 and P5, you need to show all your steps for full credit. All answers should be written in English.

P1) [3 pts] Determine an LCS of the following two strings using:

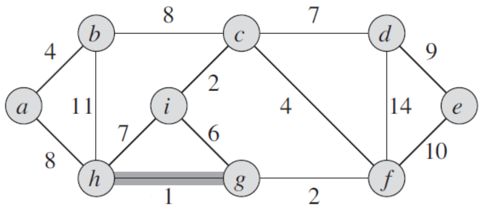
$$X = < A, B, C, B, D, A, B >, \qquad Y =< B, D, C, A, B, A >$$

Use the algorithms 'LCS-Length' and 'Print-LCS' described in the lecture note. Display the LCS using the Print-LCS algorithm.

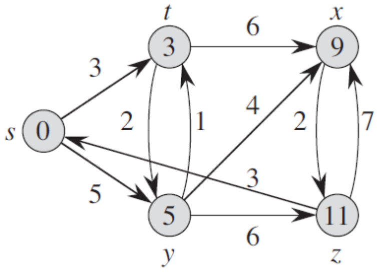
P2) [3 pts] Implement DFS to show the d and π values of each vertex on the following undirected graph, where vertex s is the source. Display the d and π values for all vertices.



P3) [3 pts] Given the following graph, run Kruskal's and Prim's algorithms and give the final minimum spanning trees. Solve the problem by hand, and show all the steps to get the final minimum spanning trees. Specifically, draw a graph for each iteration with proper coloring and the current path for each algorithm. Note that when running the Prim's algorithm, start with the vertex c.



P4) [4 pts] Given the following graph, implement and run Dijkstra's algorithm to find the shortest path from the source vertex s to vertices y and z, respectively. Display two shortest paths (with the associated vertices) and their corresponding total costs, respectively.



P5) [3 pts] Given the following table representing costs among five different cities, what is the cost of TSP if starting vertex (city) is e?

	a	b	c	d	e
a	0	30	50	20	25
b	10	0	50	35	25
c	30	50	0	10	30
d	10	20	10	0	50
e	25	25	35	50	0

Solve it by hand and show all your steps to get the full credit by referring to the example in the lecture note. First give a graph corresponding to the table and then compute the cost of TSP using dynamic programming.

P6) [4 pts] Implement and run the branch and bound algorithm to solve the 15-puzzle problem, where initial (left) and goal (right) arrangements are given as follows. You can slide four adjacent tiles into the empty space.

10	7	3	4
5	9		11
6	1	2	8
13	14	15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Display the number of moves $g(x)$, the number of misplaced tiles $h(x)$, and the total cost $\hat{C}(x)$ every move. You don't need to show bounded moves that are not selected by the algorithm.