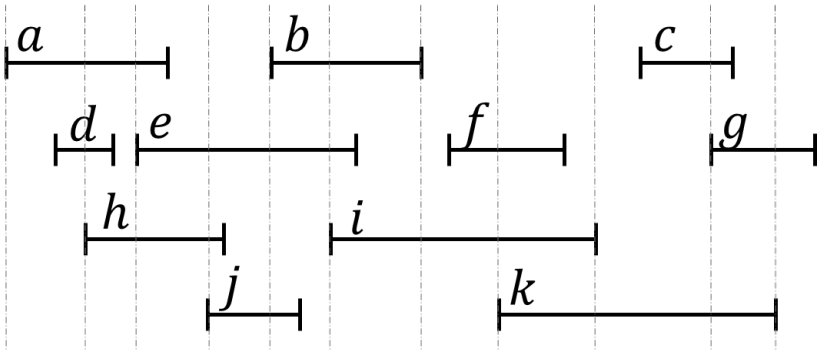
COSC 311: ALGORITHMS
MINI 6

Due Wednesday, October 17 in class

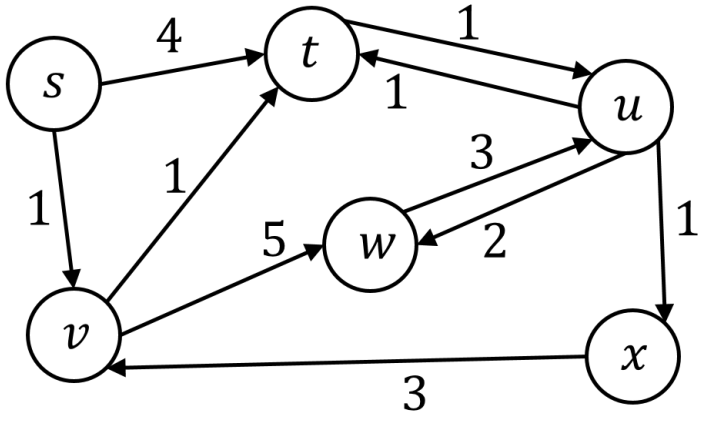
1. Interval Scheduling. Here's a set of intervals (the dashed vertical lines are to help you see where intervals start and end relative to each other):



Fill in the table to show the execution of the greedy interval scheduling algorithm discussed in class. Each row should show what the solution set, S, and the set of possible jobs to include, J, look like after adding the next job to the solution set. The first row shows, as an example, what S and J look like before the algorithm starts. (There may be more rows in the table than you need; please *leave any extra rows blank*).

After iteration	S (solution set)	J (possible jobs)
0	{} (empty set)	$a, b, c, d, e, f, g, h, i, j, k$
1	{d}	b, c, e, f, g, i, j, k
2	{d,j}	c, f, g, i, k
3	{d,j,f}	c, g
4	{d,j,f,c}	{}(empty set)
5		
6		

2. Shortest Paths. Here's a graph:



Fill in the table to show the execution of Dijkstra's algorithm on this graph to find the shortest path from node s to all other nodes. Each row should show S, the set of nodes that have been settled (including the shortest path distance from s to all nodes in S), and U, the set of nodes that are unsettled (including the best distance found so far from s to all nodes in U). The first row shows, as an example, what S and U look like after only s has been settled.

After iteration	S (solution set)	U (unsettled nodes)
1	s (dist. 0)	t (dist. 4); u (dist. ∞); v (dist. 1); w (dist. ∞); x (dist. ∞)
2	s (dist. 0); v (dist. 1);	t (dist. 2); u (dist. ∞); w (dist. 6); x (dist. ∞)
3	s (dist. 0); v (dist. 1);t (dist. 2);	u (dist. 3); w (dist. 6); x (dist. ∞)
4	s (dist. 0); v (dist. 1);t (dist. 2); u (dist. 3)	w (dist. 5); x (dist. 4)
5	s (dist. 0); v (dist. 1);t (dist. 2); u (dist. 3); x (dist. 4)	w (dist. 5)
6	s (dist. 0); v (dist. 1);t (dist. 2); u (dist. 3); x (dist. 4); w (dist. 5)	(empty)