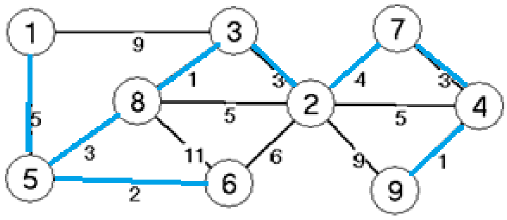
Algorithms and Data Structures II.

Name: **Example test**

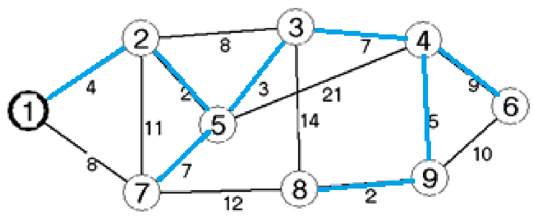
1) Illustrate the run of Prim's algorithm on the undirected graph below with vertex 1 as the source. Show the initial c and π values, and Q. Then line by line show the vertex selected for expansion, the c and π values of the vertexes, and Q after the expansion. When the algorithm is nondeterministic, consider the vertexes in alphabetical order. Draw the minimum spanning tree represented by the final π and c values.

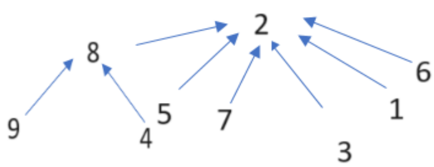


c									MST	pi								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	∞	∞	∞	∞	∞	∞	∞	∞		0	0	0	0	0	0	0	0	0
		9		5					1			1		1				
					2		3		5						5		5	
	6								6		6							
	5	1							8		8	8						
	3								3		3							
			5			4		9	2				2			2		2
			3						7				7					
								1	4									4
0	3	1	3	5	2	4	3	1		0	3	8	7	1	5	2	5	4

2) Illustrate the run of Kruskal's algorithm on the undirected graph above. Show the step-by-step transformation of the spanning forest. When the algorithm is nondeterministic, consider the vertexes and edges in alphabetical order.

	{1} {2} {3} {4} {5} {6} {7} {8} {9}
2 (2,5)	{1} {2,5} {3} {4} {6} {7} {8} {9}
2 (8,9)	{1} {2,5} {3} {4} {6} {7} {8, 9}
3 (3,5)	{1} {2,5, 3} {4} {6} {7} {8, 9}
4 (1,2)	{1, 2,3, 5} {4} {6} {7} {8, 9}
5 (4,9)	{1, 2, 3,5} {6} {7} {4, 8, 9}
7 (3,4)	{1, 2, 3, 4,5, 7, 8, 9} {6}
7 (5,7)	{1, 2, 3, 4,5,7 8, 9} {6}
8 (1,7)	
8 (2, 3)	
9 (4,6)	{1, 2, 3, 4,5,6,7, 8, 9}





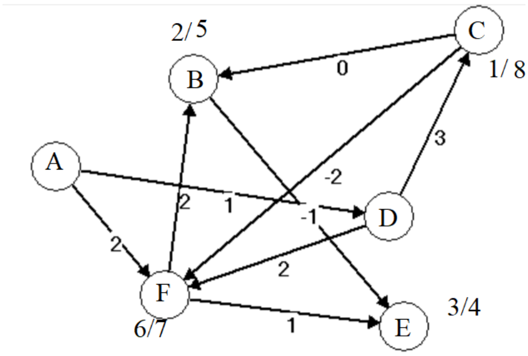
v	1		2	3		4		5		6		7		8		9	
π(v)	~~1~~	2	2	3	2	4	8	~~5~~	2	~~6~~	2	~~7~~	2	~~8~~	2	~~9~~	8
s(v)	1		~~1 2 3 4 7 8~~ 9	1		1		1		1		1		~~1 2 3~~		1	

3) Illustrate algorithm DAG-SHORTEST-PATHS on the directed graph below, using vertex 3 as the source. Draw the shortest-paths tree represented by the final π and d values.

a ⟶ d; 1; f; 2. b ⟶ e;-1. c ⟶ b; 0; f; -2.

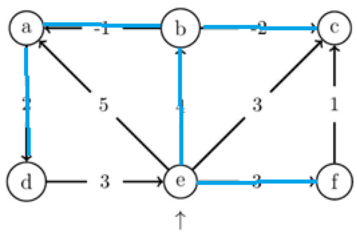
d ⟶ c; 3; f; 2. e. f ⟶ b; 2; e; 1.

d						MST	pi					
A	B	C	D	E	F		A	B	C	D	E	F
∞	∞	0	∞	∞	∞		0	0	0	0	0	0
	0				-2	C		C				C
				-1		F					F	
						B						
						E						
∞	0	0	∞	-1	-2		0	C	0	0	F	C



Topological order: <C,F,B,E>

4) Illustrate the run of the Queue-based Bellman-Ford algorithm on the directed graph below with vertex e as the source. When the algorithm is nondeterministic, consider the vertexes in alphabetical order. Show the initial d, e, π values, and Q. Then line by line show the vertex selected for expansion, the new d, e and π values of the vertexes, and Q after the expansion! Draw the shortest-paths tree represented by the final π and d values.



d						expanded vertex:d;e	Q	pi					
A	B	C	D	E	F			A	B	C	D	E	F
∞	∞	∞	∞	0;0	∞		<E>	0	0	0	0	0	0
5;1	4;1	3;1			3;1	E:0;0	<A,B,C,F>	E	E	E			E
			7;2			A:5;1	<B,C,F,D>				A		
3;2		2;2				B:4;1	<C,F,D,A>	B		B			
						C:2;2	<F,D,A>						
						F:3;1	<D,A>						
						D:7;2	<A>						
			5;3			A:3;2	<D>				A		
						D:5;3	<>						
3;2	4;1	2;2	5;3	0;0	3;1			B	E	B	A	0	E

5) We represent the computers of the network with a graph. The weights of the edges should be the probabilities of the probability that the data will pass between the two computers. Assume that these probabilities are independent. (Then the probabilities can be multiplied together, i.e. u->v->w the probability is p(u,v)*p(v,w).) Let's apply the Dijkstra algorithm to the task!

Most_reliable_paths_M(M/1:R[n,n];p/1:R[n];π/1:R[n];s:N)