

## Dijkstra's Algorithm - Adjacency Matrix Version

		adjacency matrix (represents the graph) The Start Node is node #1																
		(to nodes)																
(from nodes)		1	2	3	4	5												
1		.	50	30	100	10												
2		.	.	.	.	.												
3		.	5	.	.	.												
4		.	20	50	.	.												
5		.	.	.	10	.												
							<b>BackPoint Array</b>					<b>Dist Array</b>						
	<b>STEP</b>	<b>vertex</b>	<b>S</b>						<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
	Init	.	{1}	1	1	1	1	1	0	50	30	100	10					
	1	5	{1,5}	1	1	1	5	1	0	50	30	<b>20</b>	<del>10</del>					
	2	4	{1,4,5}	1	4	1	5	1	0	40	<b>30</b>	20	<del>10</del>					
	3	3	{1,3,4,5}	1	3	1	5	1	0	<b>35</b>	30	20	<del>10</del>					
	4	2	{1,2,3,4,5}	1	3	1	5	1	0	35	30	20	<del>10</del>					
<p style="color: blue;">In step #1 we choose vertex 5 because Dist[5]=10 is minimal. Dist[4] then changes from 100 to 20 because path 1-&gt;5-&gt;4 is shorter than 1-&gt;4. BackPoint[4] changes to 5 to indicate the new preferred path, which goes to vertex 5 just prior to ending at vertex 4.</p> <p style="color: blue;">In step #2, we choose vertex 4 because Dist[4]=20 is now minimal. Dist[2] changes to 40 because 1-&gt;5-&gt;4-&gt;2 is shorter than 1-&gt;2. BackPoint[2] changes to 4 to indicate the new preferred path to vertex 2.</p> <p style="color: blue;">In step #3 we choose vertex 3 because Dist[3]=30 is now minimal. Dist[2] changes to 35 because 1-&gt;3-&gt;2 is shorter than 1-&gt;5-&gt;4-&gt;2. BackPoint[2] changes accordingly to 3.</p>																		