

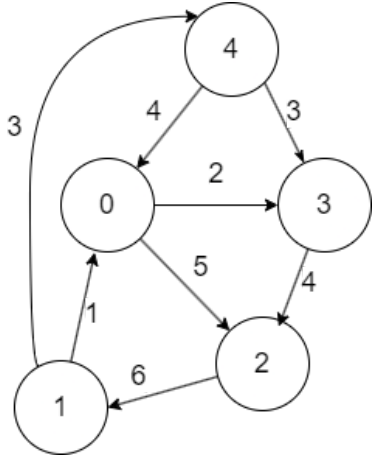
Name :	ID:	Section:
--------	-----	----------

Question 1 [15 Points]

Given a directed weighted graph, a starting vertex s and total available points p , traverse the graph, always choosing the outgoing edge with the **minimum weight at each step**. When traversing an edge with weight w , expend w points from total available points.

Determine the last vertex you can reach from s before running out of points. If you reach a **dead end (no outgoing edges)** before using all your points, that vertex is the last reachable vertex.

Note: You can solve using either adjacency matrix or adjacency list. Assume that the **Graph** is already created.

Given Graph & Sample Input	Sample Output
<p>graph =</p>  <p>$s = 0$ $p = 13$</p> <p><code>findMinPath(graph, s, p)</code></p> <p>Here, the “graph” parameter is a square Matrix if you’re using the Adjacency Matrix. Otherwise, it is an array of singly Nodes if you’re using Adjacency List.</p>	<p>Last reachable vertex = 0</p>
	<p>Explanation:</p> <p>Choosing min weight edge from the starting point $s = 0$,</p> <p>i) vertex 0 \rightarrow vertex 3 cost = 2, points available = $13 - 2 = 11$</p> <p>ii) vertex 3 \rightarrow vertex 2 cost = 4, points available = $11 - 4 = 7$</p> <p>iii) vertex 2 \rightarrow vertex 1 cost = 6, points available = $7 - 6 = 1$</p> <p>iv) vertex 1 \rightarrow vertex 0 cost = 1, points available = $1 - 1 = 0$</p> <p>Cannot go any further with available points.</p> <p>Therefore, last reachable vertex = 0</p>