

<https://www.coursera.org/learn/advanced-data-structures/lecture/38zHi/support-multiplying-adjacency-matrices>
<https://www.coursera.org/learn/advanced-data-structures/supplement/cREMI/setting-up-java-eclipse-and-the-starter-code>

**Vertex:** a node or object in a graph. The plural of vertex is "vertices".

**Edge:** a link between two vertices in the graph. Each edge is incident on two vertices. If the edge is directed, then one of these vertices is the start point and the other vertex is the endpoint. An edge may have a weight or cost associated with traversing it. **Self-loop:** an edge whose start and end points are the same vertex. **Multiple edges:** a pair of vertices may have multiple edges between them.

**In-degree:** the in-degree of a vertex,  $v$ , is the number of edges with  $v$  as their endpoint.

**Out-degree:** the out-degree of a vertex,  $v$ , is the number of edges with  $v$  as their startpoint.

**Degree:** the degree of a vertex is the sum of its in-degree and its out-degree.

**Degree sequence:** an ordered list containing the degrees of each of the vertices in a graph, in sorted order (with repetitions). The degree sequence of a graph is an invariant of the graph and can be used to distinguish between two graphs and to analyze graph properties.

**Neighbor:** a vertex,  $u$ , is a neighbor of a vertex,  $v$ , if there's an edge that's incident on both  $u$  and  $v$ . Occasionally it's useful to distinguish between the neighbors of  $v$  that are the startpoints of edges whose endpoint is  $v$  (**in-neighbors**) and those neighbors of  $v$  that are the endpoints of edges whose startpoint is  $v$  (**out-neighbors**).

**Adjacency matrix:** a representation of the edge relation in a graph. After labeling the vertices with integers, edges can be represented as nonzero entries in a 2D square array whose number of rows (and number of columns) is the number of vertices in the graph. The entry in row  $i$  and column  $j$  in the array corresponds to edge(s) with startpoint the vertex labelled  $i$  and endpoint the vertex labelled  $j$ .

**Adjacency list:** another representation of the edge relation in a graph. This data structure has a list associated with each vertex in the graph; the contents of this list are (representations of) each edge whose startpoint is that vertex.

**Path:** a path in a graph is a sequence of vertices and edges in the graph  $v_1 e_1 v_2 e_2 \dots v_i e_i \dots v_m e_m$  so that (for each  $i$ ) the startpoint of  $e_i$  is  $v_i$  and its endpoint is  $v_{i+1}$ . Paths are also referred to as walks or tours in the graph.

**Path length:** the length of a path in the graph is the number of times it traverses edges. If a single edge is traversed multiple times in a path, it gets counted each of those times. This path length doesn't take into account any weights on edges (if they are present).

**Path cost:** the total cost of traversing a path in a weighted graph is (typically) the sum of the weights of each of the edges it traverses. Depending on the meaning of the costs, the total

weights of each of the edges it traverses. Depending on the meaning of the costs, the total cost may be computed by multiplying them instead of adding.

**Distance:** the distance between two vertices in a graph is the shortest length of a path between them.

