University of Nottingham
UK | CHINA | MALAYSIA

Graph

The original slides were prepared by Dr. Jianfeng Ren.

Edited by Heshan Du

Implement the graph ADT

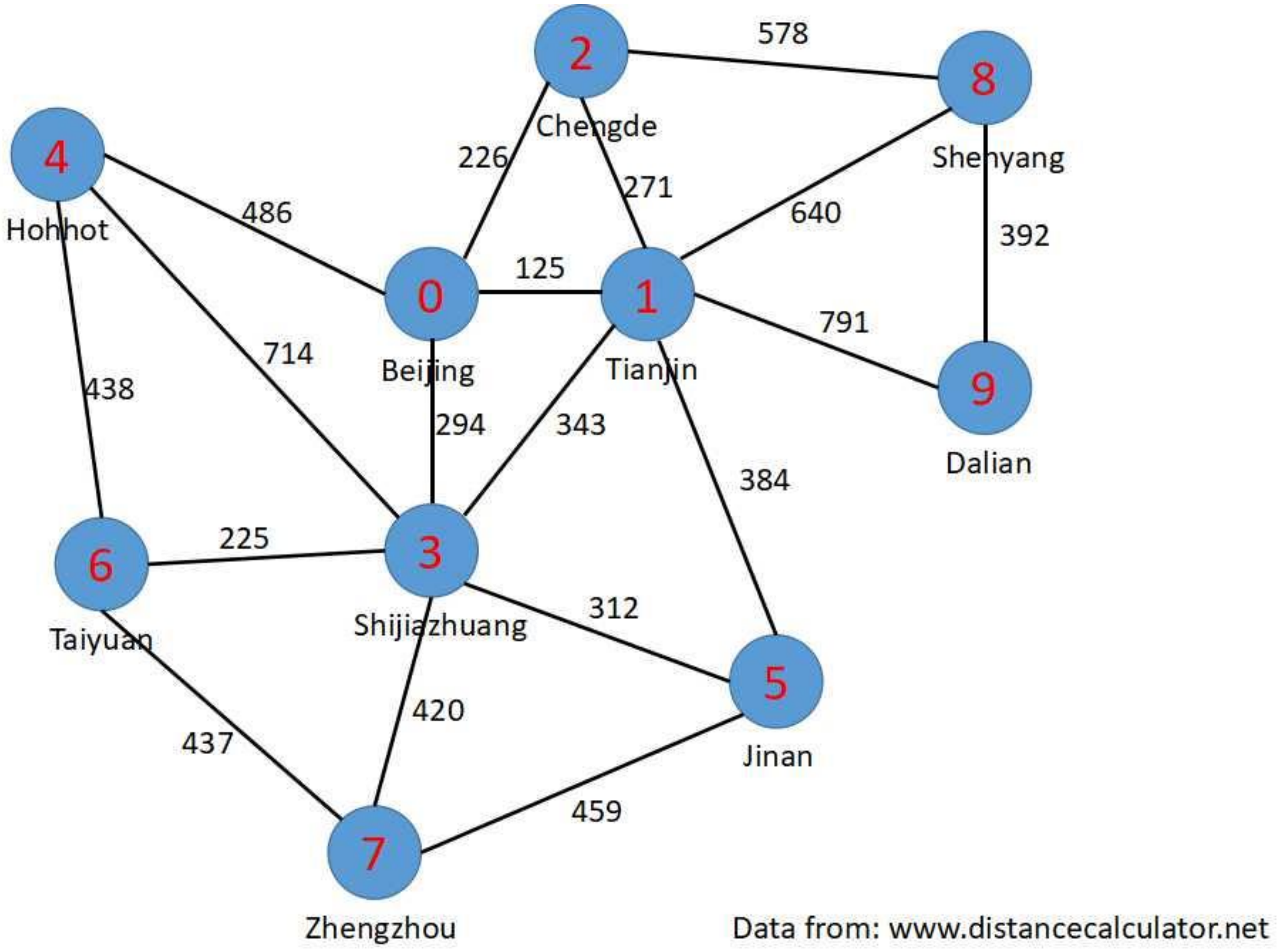
The graph ADT includes the following methods:

- **vertices( )**: Returns an iteration of all the vertices of the graph.
- **edges( )**: Returns an iteration of all the edges of the graph
- **insertVertex(x)**: Creates and returns a new Vertex storing element x.
- **insertEdge(u, v, x)**: Creates and returns a new Edge from vertex u to vertex v, storing element x; an error occurs if there already exists an edge from u to v.
- **removeVertex(v)**: Removes vertex v and all its incident edges from the graph.
- **removeEdge(e)**: Removes edge e from the graph
- ...

For more details, refer to 14.1.1 The Graph ADT.

Build the graph using adjacency matrix

Create the adjacency matrix for the following graph.



Data from: www.distancecalculator.net

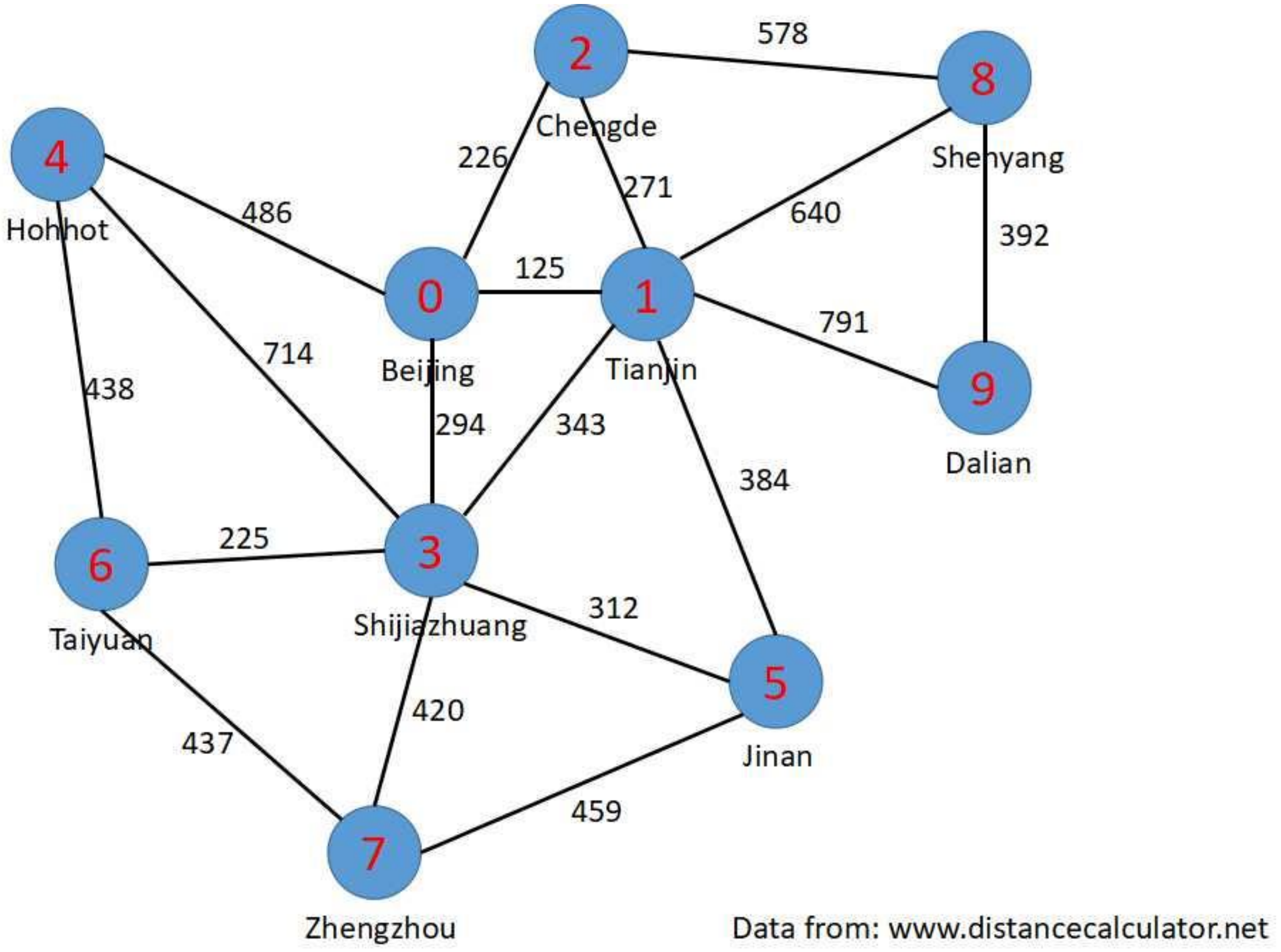
Build the graph using adjacency matrix

You should implement the graph ADT based on adjacency matrix, which includes the following methods:

- numVertices( ), numEdges( ).
- vertices( )
- edges( )
- getEdge(u, v)
- outDegree(v), inDegree(v)
- outgoingEdges(v), incomingEdges(v)
- insertVertex(x),
- insertEdge(u, v, x)
- removeEdge(e)
- removeVertex(v)

Build the graph using adjacency list

Create the adjacency list for the following graph.



Data from: www.distancecalculator.net

Build the graph using adjacency list

You should implement the graph ADT based on adjacency list, which includes the following methods:

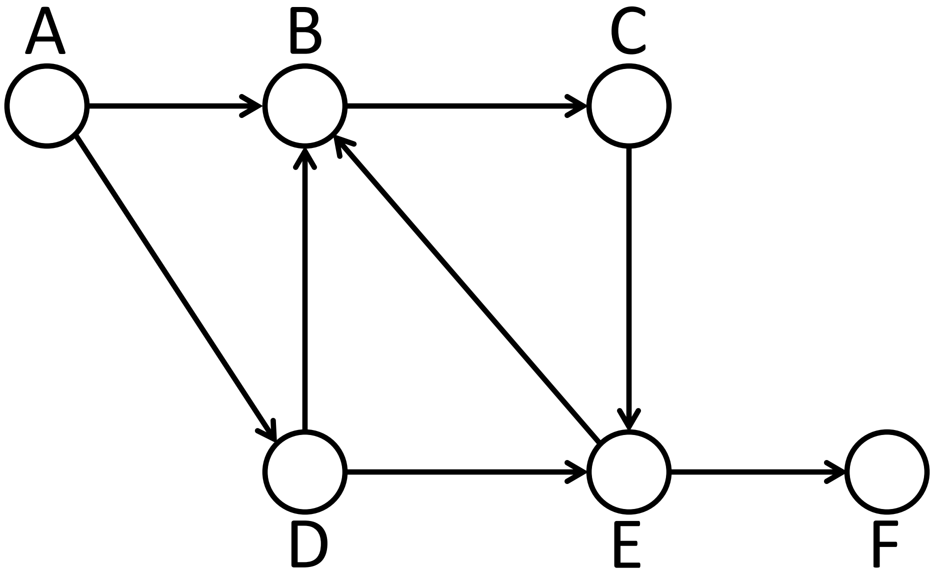
- numVertices( ), numEdges( ).
- vertices( )
- edges( )
- getEdge(u, v)
- outDegree(v), inDegree(v)
- outgoingEdges(v), incomingEdges(v)
- insertVertex(x),
- insertEdge(u, v, x)
- removeEdge(e)
- removeVertex(v)

Implement the graph using Java

There is one example of implementing the graph using an adjacency map in section 14.2.5. You may take this as reference when implementing the graph using other data structures.

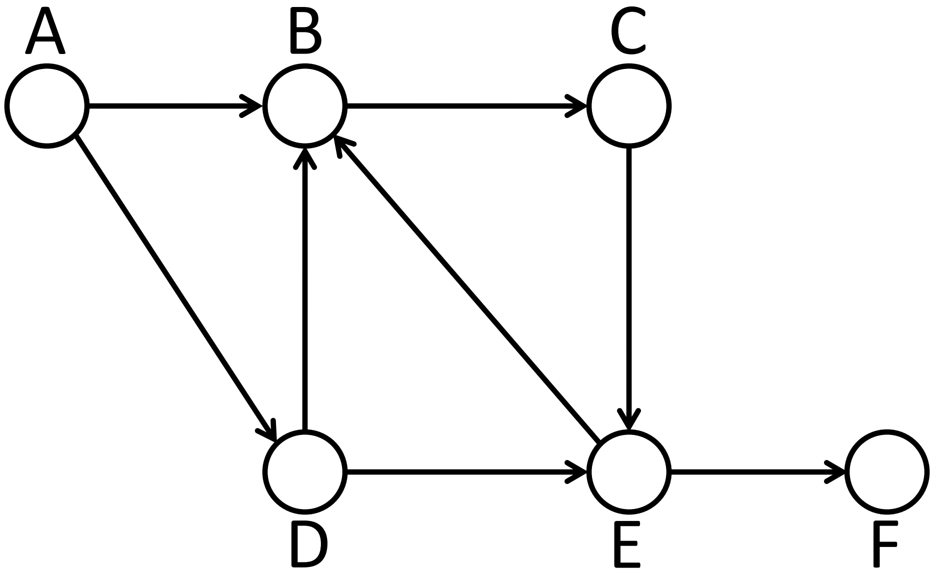
Implement BFS algorithm

- You should implement the BFS algorithm using Java. There is one example given in section 14.3.3.
- Then, use this algorithm, to find the traversal order for the following graph (assume starting from A).



Implement DFS algorithm

- You should implement the DFS algorithm using Java. There is one example given in section 14.3.2.
- Then, use this algorithm, to find the traversal order for the following graph (assume starting from A).



Implement the Dijkstra’s algorithm

- You should implement the Dijkstra’s algorithm using Java.
- There is one example given in Section 14.6.2: Programming Dijkstra's Algorithm in Java. (CF 14.13)
- There is another example (CF 14.14), which reconstructs a single-source shortest-path tree, based on knowledge of the shortest-path distance.

Implement topological sorting

- Implement the topological sorting algorithm.
  - ➢There is a sample implementation in Section 14.5.1, CF 14.11.
  - ➢Use the algorithm to find the topological order for the graph.
- Implement the modified topological sorting algorithm to detect cycle.
  - ➢When no vertex can be removed, but some vertices remain in the graph, there is a cycle.

Implement the Prim's algorithm

- Implement the Prim's algorithm.
  - ➢Use the algorithm to find the MST.
  - ➢There is an implementation for Kruskal's algorithm for MST, in section 14.7.2, CF 14.17. You may use it as a reference.