Adjacency Matrix

Adjacency List

Public Edge first(int v)

Public Edge next(Edge w)

0 1 2 3 4

0 1 0 0 1 0

0 0 0 1 0 1

0 0 0 0 1 2

0 0 1 0 0 3

0 1 0 0 0 4

Next(edge(0, 1)) → edge(0, 4)

next(edge(0,4)) → null

Access all eddges by using the two methods in a nested for loop.

Public void setEdge(int i, int j, int weight) //for directed graph

{

assert(weight != 0) : “Cannot set weight to zero”;

if(matrix[i][j] == 0) numEdge++;

Matrix[i][j] = weight;

}

//this method used to either insert new edge into graph or update weight of existing edge.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| **0** |  | 1 |  |  | 1 |
| **1** | 1 |  |  | 1 | 1 |
| **2** |  |  |  | 1 | 1 |
| **3** |  | 1 | 1 |  |  |
| **4** | 1 | 1 | 1 |  |  |

Undirected graph

EDGE 3,2 is same as 2,3 in undirected graph. To change setEdge for undirected graph do the following.

Public void setEdge(int i, int j, int weight) //for UNDIRECTED graph

{

assert(weight != 0) : “Cannot set weight to zero”;

if(matrix[i][j] == 0 && matrix [j][i] == 0) numEdge++;

Matrix[i][j] = weight;

Matrix[j][i] = weight;

}

-------------------------

How to build following graph in main



(Numbers on lines are weights)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| **0** | 0 | 1 | 0 | 0 | 1 |
| **1** | 0 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 |
| **3** | 0 | 0 | 1 | 0 | 0 |
| **4** | 0 | 1 | 0 | 0 | 0 |

Main()

{

Graph g = new Graph(5)

g.setEdge(0, 1, 25);

g.setEdge(0, 4, 31);

g.setEdge(1, 3, 11);

}

We may need to overload setEdge to account for not caring about weight (so setEdge(int i, int j))

**Graph traversal**

-Some applications requirses visiting every vector in a graph exactly

Focus on first connected Graphs( No Island)

* Two types of Traversals
  + Depth First Search (DFS)
  + Breadth First Search (BFS)