DATA STRUCTURE AND ALGORITHM

CLASS 8

Seongjin Lee

Updated: 2017-03-06
DSA_2017_08

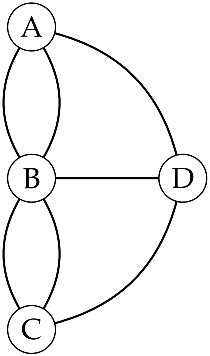
insight@gnu.ac.kr
http://resourceful.github.io
Systems Research Lab.
GNU

Table of contents

Graph

First use of Graph

- Euler used graph to solve the Koenigsberg bridge problem in 1736
- "Starting at some land area, is it possible to return to the starting point after walking across each of the bridges exactly once?"

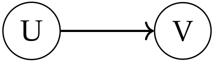


Definitions and Notations

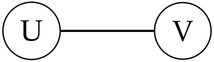
- **Notation**: $G = (V, E)$
- V is the vertex set. Ex. $V = \{0, 1, 2, 3\}$
  - Vertices are also called nodes and points
- E is the edge set. Ex. $E = \{(0, 1), (1, 3), (2, 3), (0, 2)\}$
  - Each edge connects two different vertices
  - Edges are also called acrs and lines

Directed and Undirected Graphs

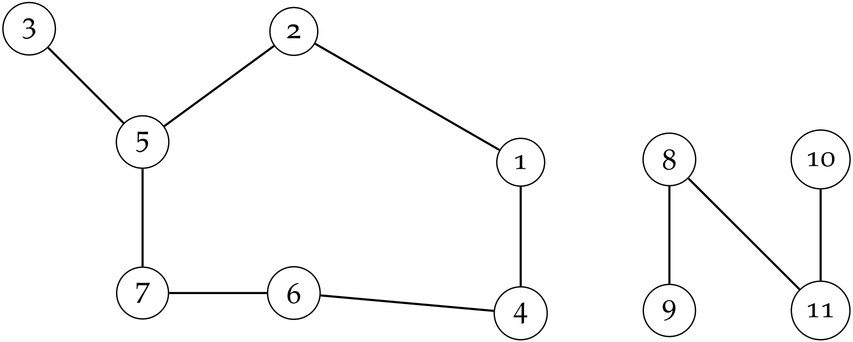
- Directed Edge has an *Orientation* (u, v)
  - Directed graph = Every Edge has an orientation



- Undirected edge has no *Orientation* (u, v)
  - Undirected graph = No oriented edge

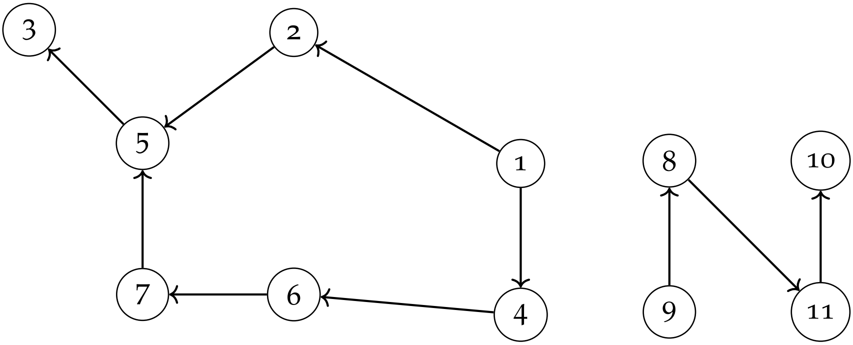


Undirected Graph

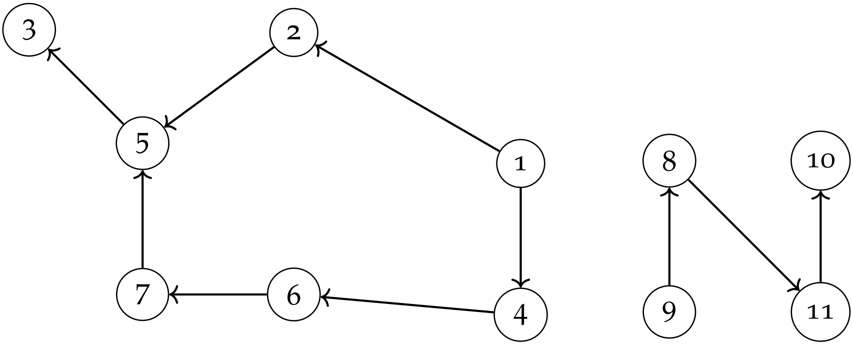


- $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$
- $E = \{(1, 2), (1, 4), (2, 5), (4, 6), (3, 5), (5, 7), (6, 7), (8, 9), (8, 11), (10, 11)\}$

Directed Graph



Directed Graph

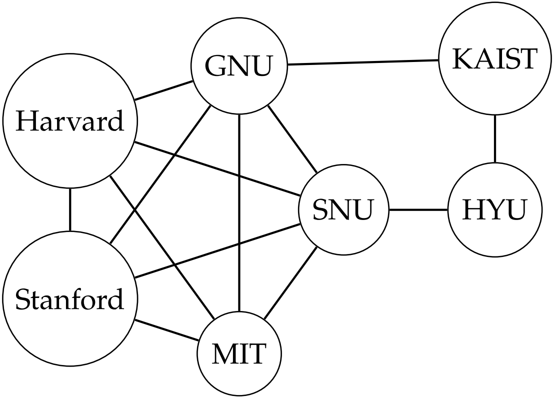


- $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$
- $E = \{< 1, 2 >, < 1, 4 >, < 2, 5 >, < 4, 6 >, < 5, 3 >, < 6, 7 >, < 7, 5 >$ $, < 8, 9 >, < 8, 11 >, < 11, 10 >\}$

Applications

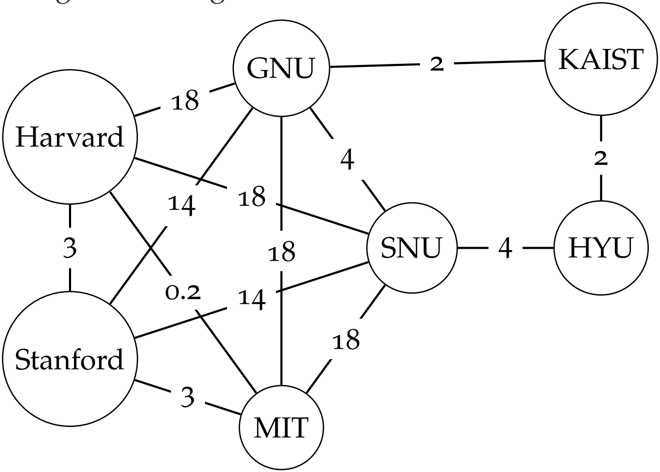
Communication Network

- Vertex = University
- Edge = Communication link



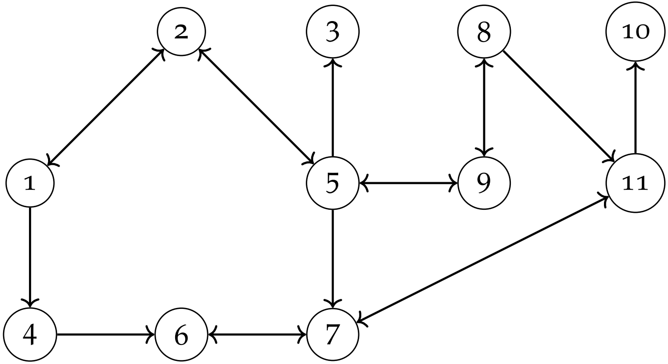
Driving Distance/Time Map

- Vertex = City
- Edge Weight = driving distance or time



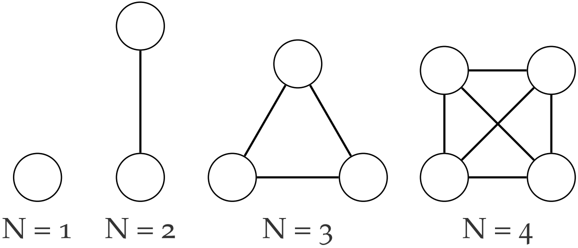
Applications

Some streets are one way



Complete Undirected Graph

- A graph that has the maximum number of edges
- A graph that has all possible edges



Number of Edges in Undirected Graph

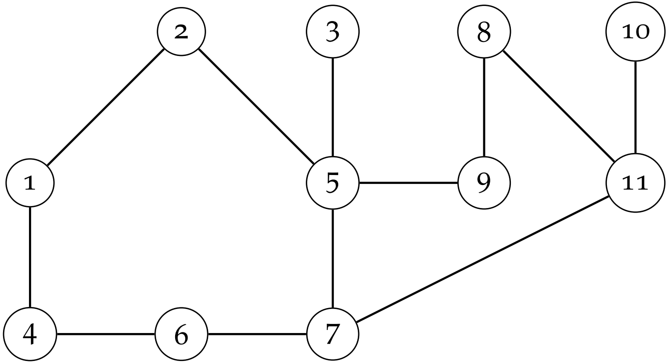
- Each edge is of the form (u, v), and $u \neq v$
- Number of such pairs in an n vertex graph is $n(n-1)$.
- Since edge (u, v) is the same as edge (v, u), the number of edges in a complete undirected graph is $\frac{n(n-1)}{2}$.
- Number of edges in an undirected graph is $\leq \frac{n(n-1)}{2}$.

Number of Edges in Directed Graph

- Each edge is of the form $< u, v >$, $u \neq v$.
- Number of such pairs in an n vertex graph is $n(n-1)$.
- Since edge $< u, v >$ is not the same as edge $< v, u >$, the number of edges in a complete directed graph is $n(n-1)$.
- Number of edges in a directed graph is $<= n(n-1)$.

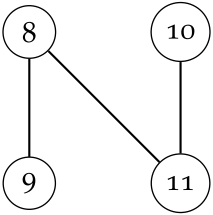
Vertex Degree

- **Vertex Degree** is the number of edges of the vertex
- Degree(2) = 2, Degree(5) =3, Degree(3) = 1,



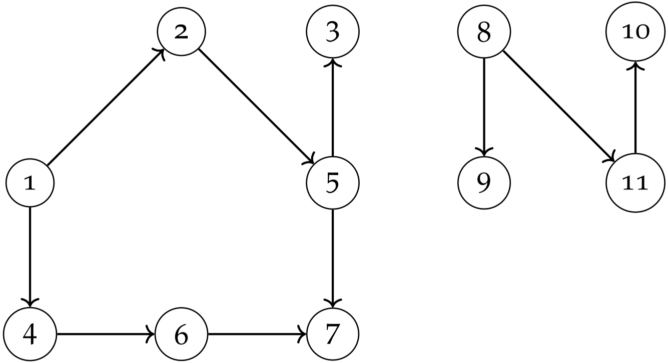
Sum of Vertex Degrees

- **Sum of Vertex Degrees** is $2 \times e$, where e is number of edges



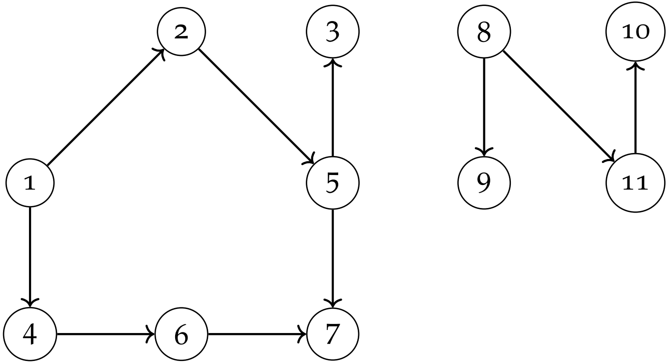
In-Degree of a Vertex

- In-degree of a vertex is the number of incoming edges
- In-degree(2) = 1, in=degree(8) = 0



Out-Degree of a Vertex

- Out-degree of a vertex is the number of outbound edges
- Out-degree(2) = 1, out=degree(8) = 2



Sum of In- and Out-Degrees

- Each edge contributes 1 to the in-degree of some vertex and 1 to the out-degree of some other vertex
- Sum of in-degree = sum of out-degree = e, where e is the number of edges in the directed graph

Other Definitions

- **Graph**: $G = (V, E)$
- If G is undirected graph and (u, v) is an edge of G, then
  - u and v are **adjacent**
  - (u, v) is **incident** on vertices u and v
- If G is directed graph and $< u, v >$ is an edge of G, then
  - u is **adjacent** to v
  - v is **adjacent** from u
- Path from u to v
- **Simple path** : a path in which all vertices, except possibly the first and the last, are distinct
- **Cycle** : a simple path in which the first and the last vertices are the same

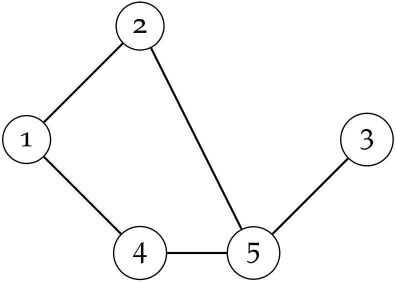
Graph Representation

Graph Representation

- Adjacency Matrix
- Adjacency Lists
  - Linked Adjacency Lists
  - Array Adjacency Lists

Adjacency Matrix

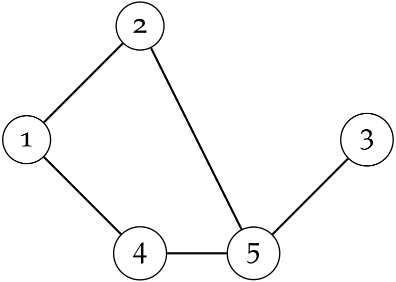
- $n \times n$ matrix with 0 and 1 representing the incidents ,where n = # of vertices
- $A(i, j) = 1$ iff (i, j) is an edge



	1	2	3	4	5
1	0	1	0	1	0
2	1	0	0	0	1
3	0	0	0	0	1
4	1	0	0	0	1
5	0	1	1	1	0

Propertiex of Adjacency Matrix

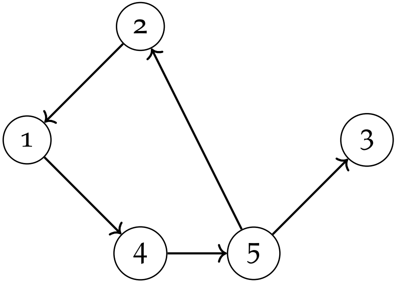
- Diagonal entries are zero
- Adjacency matrix of an undirected graph is symmetric
  - $A(i,j) = A(j,i)$ for all i and j



	1	2	3	4	5
1	0	1	0	1	0
2	1	0	0	0	1
3	0	0	0	0	1
4	1	0	0	0	1
5	0	1	1	1	0

Adjacency Matrix for Directed Graph

- Diagonal entries are zero
- Adjacency matrix of a directed graph need not be symmetric



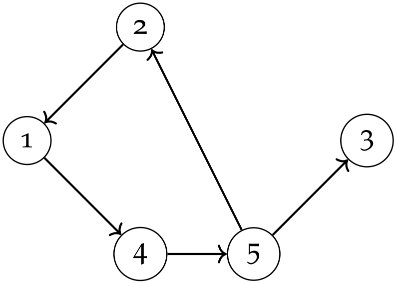
	1	2	3	4	5
1	0	1	0	1	0
2	1	0	0	0	1
3	0	0	0	0	1
4	1	0	0	0	1
5	0	1	1	1	0

Properties of Adjacency Matrix

- n^2 bits of space is required
- For an undirected graph, may store only lower or upper triangle (exclude the diagonal)
  - $\frac{n(n-1)}{2}\ bits$
- $O(n)$ time to find vertex degree and/or vertices adjacent to a give vertex

Adjacency Lists

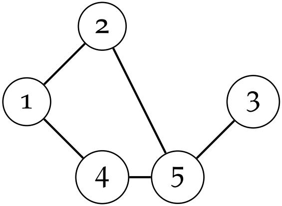
- Adjacency list for vertex i is a linear list of vertices adjacent from vertex i.
- An array of n adjacency lists.

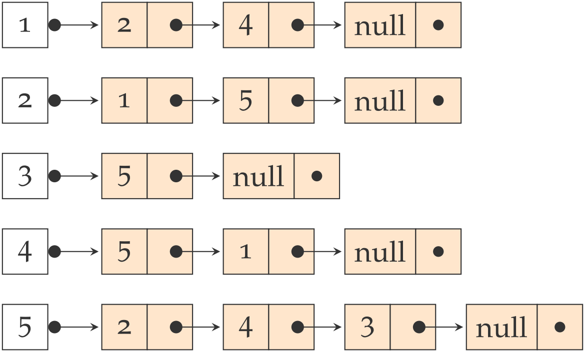


AdjList[1] = (2, 4)
AdjList[2] = (1, 5)
AdjList[3] = (5)
AdjList[4] = (5, 1)
AdjList[5] = (2, 4, 3)

Linked Adjacency Lists

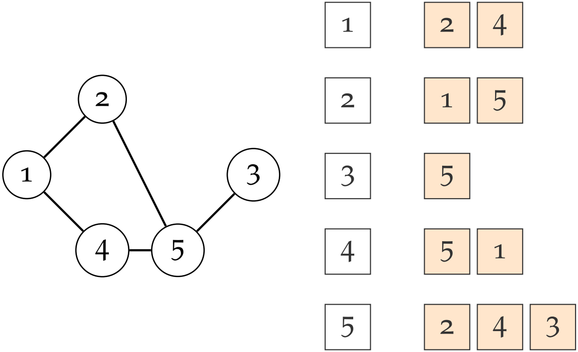
- Each adjacency list is a chain
  - Array length = n
  - # of chain nodes = $2e$ (undirected graph)
  - # of chain nodes = e (directed graph)





Linked Adjacency Lists

- Each adjacency list is an array list
  - Array length = n
  - # of chain nodes = $2e$ (undirected graph)
  - # of chain nodes = e (directed graph)



Weighted Graphs

- Cost Adjacency Matrix
  - $C(i, j)$ = cost of edge (i, j)
- Each list element of adjacency lists is a pair `(Adjacent vertex, Edge weight)`