

CS213/293 Data Structure and Algorithms 2024

Lecture 12: Graphs - basics

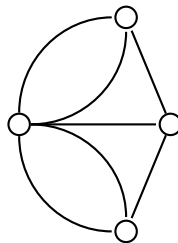
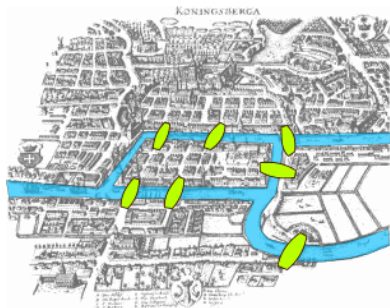
Instructor: Ashutosh Gupta

IITB India

Compile date: 2024-10-13

The problem of Königsberg's bridges

Problem: find a walk through the city that would cross each of those bridges once and only once.



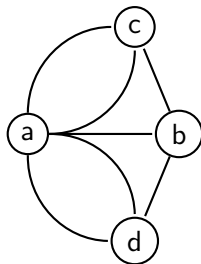
We may view the problem as visiting all nodes without repeating an edge in the above graph.

(Source: Wikipedia)

The first graph theory problem. Euler gave the solution!

Graphs

A **graph** has **vertices** (also known as nodes) and vertices are connected via **edges**.



The above is a graph $G = (V, E)$, where

$V = \{a, b, c, d\}$ and

E is a multiset.

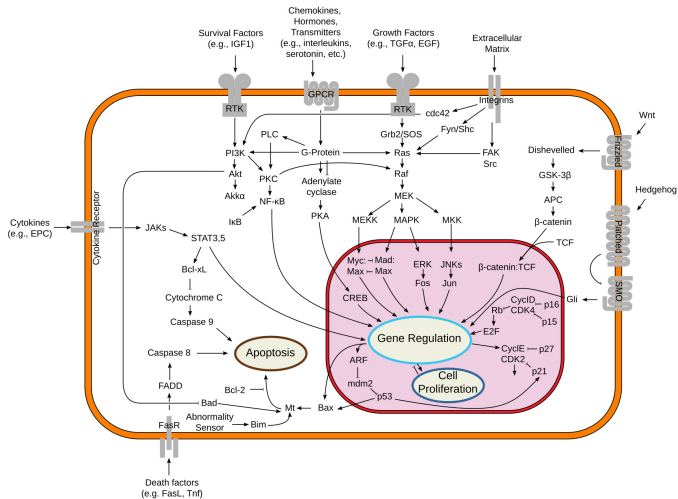
$E = \{\{a, b\}, \{a, c\}, \{a, c\}, \{a, d\}, \{a, d\}, \{b, c\}, \{b, d\}\}$.

Example: graphs are everywhere



(Source: Internet)

Example: graphs are everywhere (2)



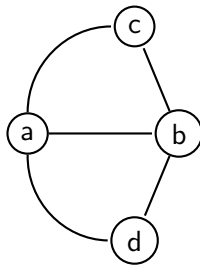
(Source: Wikipedia)

Formal definition

Definition 12.1

A graph $G = (V, E)$ consists of

- ▶ set of vertices V and
- ▶ set of edges E is a set of unordered pairs of elements of V .



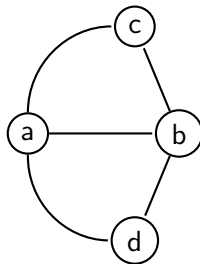
Commentary: In the bridge example, E was a multiset and here E is a set. If we want to support multiset, we can define $E \subseteq \text{unorderedPairs}(V) \times \mathbb{N}$, which is a natural extension of the above definition. $\text{unorderedPairs}(V) = \{\{a, b\} | a, b \in V \wedge a \neq b\}$

Topic 12.1

Basic Terminology

Adjacency and degree

Example 12.1



$adjacent(a) = \{c, b, d\}$ and $adjacent(d) = \{a, b\}$.

$degree(a) = 3$ and $degree(d) = 2$.

Consider a graph $G = (V, E)$.

Definition 12.2

Let $adjacent(v) = \{v' | \{v, v'\} \in E\}$.

Definition 12.3

Let $degree(v) = |adjacent(v)|$.

Exercise 12.1

- What is $\sum_{v \in V} degree(v)$?
- Is $\{v, v\} \in E$ possible?

Commentary: $\sum_{v \in V} degree(v) = 2|E|$

Paths, simple paths, and cycles

Consider a graph $G = (V, E)$.

Definition 12.4

A **path** is a sequence of vertices v_1, \dots, v_n such that $\{v_i, v_{i+1}\} \in E$ for each $i \in [1, n)$.

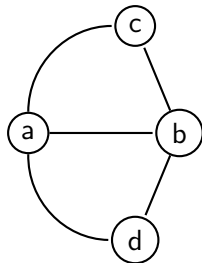
Definition 12.5

A **simple path** is a path v_1, \dots, v_n such that $v_i \neq v_j$ for each $i < j \in [1, n]$.

Definition 12.6

A **cycle** is a path v_1, \dots, v_n such that v_1, \dots, v_{n-1} is a simple path and $v_1 = v_n$.

Example 12.2



$abcad$ is a path but not a simple path.

abd is a simple path.

$abda$ is a cycle.

Exercise 12.2

a. Can there be an empty path?

b. Is b a cycle?

Subgraph

Consider a graph $G = (V, E)$.

Definition 12.7

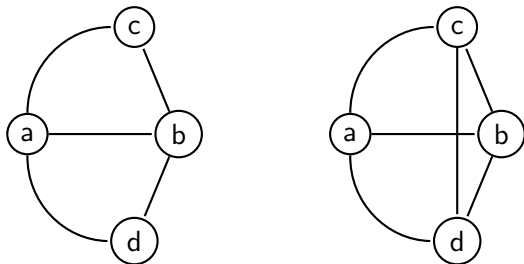
A graph $G' = (V', E')$ is a **subgraph** of G if $V' \subseteq V$ and $E' \subseteq E$.

Definition 12.8

For a set of vertices V' , let $G - V'$ be $(V - V', \{e | e \in E \wedge e \subseteq V - V'\})$.

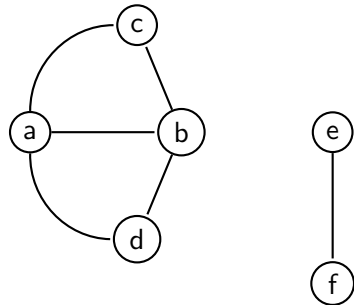
Example 12.3

The left graph is a subgraph of the right graph.



Connected graph

Example 12.4



The above is not a connected graph.

The above has two connected components.

Consider a graph $G = (V, E)$.

Definition 12.9

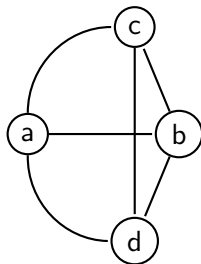
G is **connected** if for each $v, v' \in V$ there is a path v, \dots, v' in E .

Definition 12.10

A graph G' is a **connected component** of G if G' is a maximal connected subgraph of G .

Complete graph

Example 12.5



Exercise 12.3

If $|V| = n$, how many edges does a complete graph have?

Consider a graph $G = (V, E)$.

Definition 12.11

G is a **complete graph** if for all pairs

$v_1, v_2 \in V$

- ▶ if $v_1 \neq v_2$, $v_1 \in \text{adjacent}(v_2)$, and
- ▶ if $v_1 = v_2$, $v_1 \notin \text{adjacent}(v_1)$.

Topic 12.2

Tree (a new non-recursive definition of tree)

Tree

Consider a graph $G = (V, E)$.

Definition 12.12

G is a **tree** if G is connected and has no cycles.

Definition 12.13

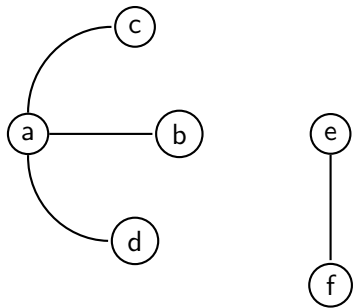
G is a **forest** if G is a disjoint union of trees.

Definition 12.14

$G = (V, E, v)$ is a **rooted tree** if (V, E) is a tree and $v \in V$ is called root.

The trees in the earlier lectures are rooted trees.

Example 12.6



The above is a forest containing two trees.

Exercise 12.4

Which nodes of a tree can be selected for root?

Every tree has a leaf

Definition 12.15

For a tree $G = (V, E)$, a node $v \in V$ is a leaf if $\text{degree}(v) \leq 1$. one side connected or not connected is leaf.

Theorem 12.1

For a finite tree $G = (V, E)$ and $|V| > 1$, there is $v \in V$ such that $\text{degree}(v) = 1$.

Proof.

Since there are no cycles in G and G is finite, there is a simple path v_1, \dots, v_n of G that cannot be extended at either end.

Therefore, there must be two nodes such that $\text{degree}(v) = 1$.



Number of edges in a tree

Theorem 12.2

For a finite tree $G = (V, E)$, $|E| = |V| - 1$.

Proof.

Base case:

Let $|V| = 2$. We have $|E| = 1$.

Induction step:

Let $|V| = n + 1$.

Consider a leaf $v \in V$ and $\{v, v'\} \in E$. Every leaf removal reduces the edge by one.

Since $\text{degree}(v) = 1$ in G , $G - \{v\}$ is a tree.

Due to the induction hypothesis, $G - \{v\}$ has $|V| - 2$ edges.

Hence proved. □

Number of edges in a tree

Theorem 12.3

Let $G = (V, E)$ be a finite graph. If $|E| < |V| - 1$, G is not connected.

Proof.

Let us suppose there are cycles in the graph.

If we remove an edge from a cycle, it does not change the connectedness of any pairs of vertices.

(Why?)

We keep removing such edges until no more cycles left.

Since $|E| < |V| - 1$, the remaining graph is not a tree. Therefore, G was not connected. □

Commentary: Please check the definition of a tree. The last conclusion is a direct application of the contra-positive of the definition of tree.

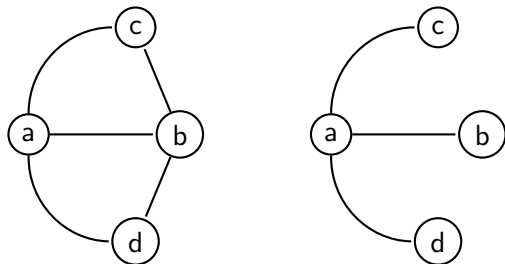
Spanning tree

Example 12.7

Consider a graph $G = (V, E)$.

Definition 12.16

A *spanning tree of G* is a subgraph of G that is a *tree and contains all vertices of G* .

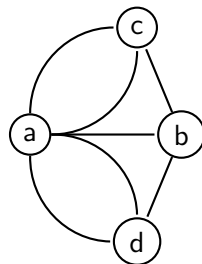


The right graph is the spanning tree of the left graph.

Topic 12.3

Multi-graph

Multi graph



Definition 12.17

A graph $G = (V, E)$ consists of

- ▶ set of vertices V and
- ▶ set of edges E is a multiset of unordered pairs of elements of V .

The above is a graph $G = (V, E)$, where

$$V = \{a, b, c, d\} \text{ and}$$

$$E = \{\{a, b\}, \{a, c\}, \{a, c\}, \{a, d\}, \{a, d\}, \{b, c\}, \{b, d\}\}.$$

Eulerian tour

Consider a graph $G = (V, E)$.

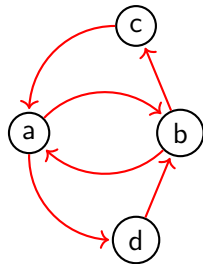
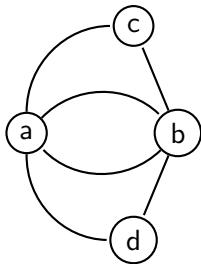
Definition 12.18

For a (multi)graph G , an **Eulerian tour** is a path that traverses every edge exactly once and returns to the same node.

Exercise 12.5

Why an Eulerian tour is not a cycle?

Example 12.8



Eulerean path: cadbabc

Theorem 12.4

A graph has an Eulerian tour if and only if all vertices have even degrees.

Because for every incoming edge we need an outgoing edge.

Proof.

Hint: Replace edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ by $\{v_1, v_3\}$.



Topic 12.4

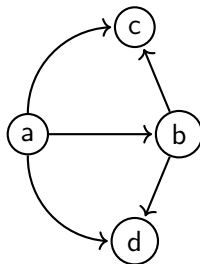
Directed graph

Directed graph

Definition 12.19

A graph $G = (V, E)$ consists of

- ▶ set of vertices V and
- ▶ set of edges $E \subseteq V \times V$.



The above is a directed graph $G = (V, E)$, where

$V = \{a, b, c, d\}$ and

$E = \{(a, b), (a, c), (a, d), (b, c), (b, d)\}$.

There is a path from a to d , but not d to a .

Definition 12.20

A **path** is a sequence of vertices v_1, \dots, v_n such that $(v_i, v_{i+1}) \in E$ for each $i \in [1, n)$.

Strongly connected component (SCC)

Consider a directed graph $G = (V, E)$.

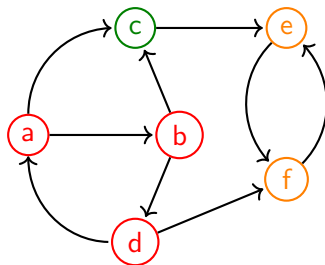
Example 12.9

Definition 12.21

G is *strongly connected* if for each $v, v' \in V$ there is a path v, \dots, v' in E .

Definition 12.22

A graph G' is a *strongly connected component (SCC)* of G if G' is a maximal strongly connected subgraph of G .



a**b****d**, **c**, and **e****f** are SCCs.

SCC-Graph

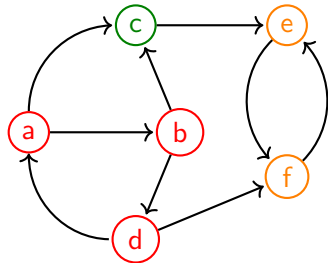
Let G be a directed graph.

Definition 12.23

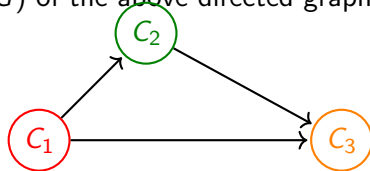
SCC-graph $SCC(G)$ is defined as follows.

- ▶ Let C_1, \dots, C_n be SCCs of G .
- ▶ For each C_i , create a vertex v_i in $SCC(G)$.
- ▶ Add an edge (v_i, v_j) to $SCC(G)$, if there are two vertices u_i and u_j in G with $u_i \in C_i, u_j \in C_j$ and $(u_i, u_j) \in E$.

Example 12.10



$SCC(G)$ of the above directed graph G is



SCC(G) is acyclic

Theorem 12.5

condensation of SCC

For any directed graph $G = (V, E)$, $SCC(G)$ is acyclic.

Proof.

Let us suppose there is a cycle in $SCC(G) = (V', E')$.

There must be $u, u' \in V'$ such that there are paths from u to u' and in the reverse direction.

Let C and C' be the SSCs in G corresponding to u and u' respectively.

There must be a path from nodes in C to nodes in C' and in the reverse direction.

C and C' cannot be SSCs of G . **Contradiction.**



Topic 12.5

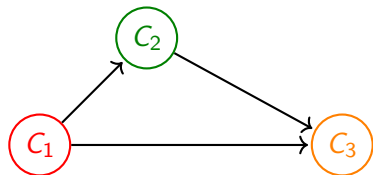
Directed acyclic graph (DAG)

Directed acyclic graph (DAG)

Consider a directed graph $G = (V, E)$.

Definition 12.24

G is a *directed acyclic graph (DAG)* if G has no cycles.



The above is a directed acyclic graph.

Exercise 12.6

Define a tree from DAG.

Commentary: We may view that DAG $SCC(G)$ is embedded in graph G .

Topic 12.6

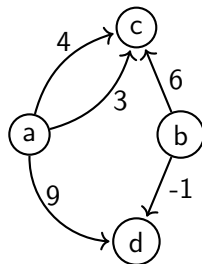
Labeled graph

Directed labeled graph

Definition 12.25

A graph $G = (V, E)$ is consists of

- ▶ set of vertices V and
- ▶ set of edges $E \subseteq V \times L \times V$,
where L is the set of labels.



The above is a labelled directed graph $G = (V, E)$, where

$L = \mathbb{Z}$, $V = \{a, b, c, d\}$ and

$E = \{(a, 3, c), (a, 4, c), (a, 9, d), (b, 6, c), (b, -1, d)\}$.

Topic 12.7

Representation of graph

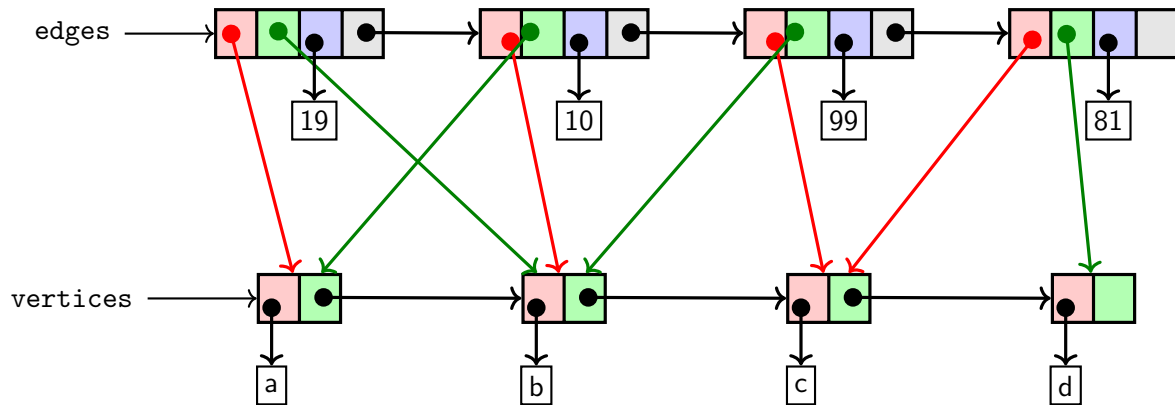
Representations of graph

- ▶ Edge list
- ▶ Adjacency list
- ▶ Matrix

Edge list

- ▶ Store vertices as a sequence (array/list)
- ▶ Store edges as a sequence with pointers to vertices

Example: edge list

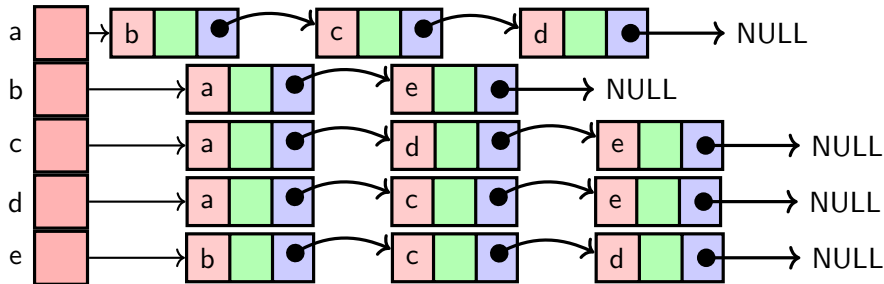


Exercise 12.7

- What is the cost of computing $\text{adjacent}(v)$?
- What is the cost of insertion of an edge?

Adjacency list

- Each vertex maintains the list of adjacent nodes.



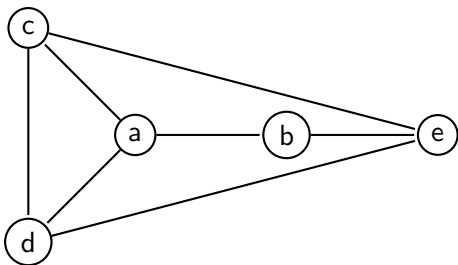
Space: $O(|V| + \sum \text{degree}(v)) = O(|V| + |E|)$

Exercise 12.8

- Draw the graph for the above data structure.
- What is the cost of $\text{adjacent}(v)$, and find vertices of an edge given by edge number?
- How can we mix the edge list and adjacency list to make the above operations efficient?

Adjacency Matrix

Store adjacency relation on a matrix.



Commentary: The matrix is stored in a multidimensional array. If we store the matrix as a vector of vectors, then it is *similar* to the adjacency list storage of the graph instead of the adjacency matrix. Many graph algorithms are sequences of matrix operations over adjacency matrices. Matrix operations are not fast on vectors of vectors.

	a	b	c	d	e
a	0	1	1	1	0
b	1	0	0	0	1
c	1	0	0	1	1
d	1	0	1	0	1
e	0	1	1	1	0

Space: $O(|V|^2)$

Exercise 12.9

- a. What is the cost of adding a node? $O(n^2)$
- b. What is the cost of `adjacent(v)`? $O(n)$
- c. What is the cost of finding vertices of an edge which is given as a pair of positions? $O(1)$
- d. How can we mix edge list and adjacency matrix?

Topic 12.8

Tutorial problems

Exercise: modeling COVID

Exercise 12.10

The graph is an extremely useful modeling tool. Here is how a Covid tracing tool might work. Let V be the set of all persons. We say (p,q) is an edge (i) in E_1 if their names appear on the same webpage, and (ii) in E_2 if they have been together in a common location for more than 20 minutes. What significance do the connected components in these graphs and what does the BFS do? Does the second graph have epidemiological significance? If so, what? If not, how would you improve the graph structure to get a sharper epidemiological meaning?

Exercise: Bipartite graphs

Definition 12.26

A graph $G = (V, E)$ is bipartite if $V = V_1 \uplus V_2$ and for all $e \in E$ $e \not\subseteq V_1$ and $e \subseteq V_2$.

Exercise 12.11

Show that a bipartite graph does not contain cycles of odd length.

Exercise: Planer graphs

Exercise 12.12

Let us take a plane paper and draw circles and infinite lines to divide the plane into various pieces. There is an edge (p,q) between two pieces if they share a common boundary of intersection (which is more than a point). Is this graph bipartite? Under what conditions is it bipartite?

Exercise: Die hard puzzle

Exercise 12.13

There are three containers A, B, and C, with capacities of 5, 3, and 2 liters respectively. We begin with A has 5 liters of milk and B and C are empty. There are no other measuring instruments. A buyer wants 4 liters of milk. Can you dispense this? Model this as a graph problem with the vertex set V as the set of configurations $c=(c_1, c_2, c_3)$ and an edge from c to d if d is reachable from c . Begin with $(5, 0, 0)$. Is this graph directed or undirected? Is it adequate to model the question: How to dispense 4 liters?

Topic 12.9

Problems


Exercise: Modeling call center

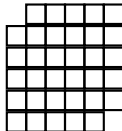
Exercise 12.14

Suppose that there are M workers in a call center for a travel service that gives travel directions within a city. It provides services for N cities - C_1, \dots, C_N . Not all workers are familiar with all cities. The numbers of requests from cities per hour are R_1, \dots, R_N . A worker can handle K calls per hour. Is the number of workers sufficient to address the demand? How would you model this problem? Assume that R_1, \dots, R_N , and K are small numbers.

Exercise: tiling (2023 Quiz)

Exercise 12.15

Prove that it is not possible to tile the following floor using some number of tiles shaped . Tiles must not be deformed and overlap.



End of Lecture 12