Principles of Computer Science II

Introduction to Graph Theory

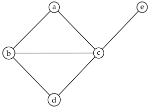
Ioannis Chatzigiannakis

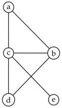
Sapienza University of Rome

Lecture 18

Graph Definition

- We denote a graph by $G = G(V, E)$, where
  - V represents the set of vertices
    $V = \{a, b, c, d, e\}$
  - E represents the set of edges
    $E = \{(a, b), (a, c), (b, c), (b, d), (c, d), (c, e)\}$





Basic Definitions

- We denote $|V| = n$ – the number of vertices.
- We denote $|E| = m$ – the number of edges.
- Two vertices u, v are called adjacent or neighboring vertices if there exists an edge $e = (u, v)$.
- We say that edge e is incident to vertices u and v.
- We say that vertices u and v are incident to edge e.
- A loop is an edge from a node to itself: (u, u).

Degree of the Vertex

- The number of edges incident to a given vertex v is called the degree of the vertex and is denoted $d(v)$.
- For every graph $G = G(V, E)$,

$$\sum_{u \in V} d(u) = 2 \cdot |m|$$

- Notice that an edge connecting vertices v and w is counted in the sum twice: first in the term $d(v)$ and again in the term $d(w)$.

Subgraphs

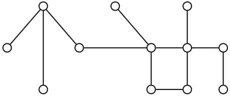
- A subgraph G' of G consists of a subset of V and E. That is, $G' = (V', E')$ where $V' \subset V$ and $E' \subset E$.
- A spanning subgraph contains all the nodes of the original graph.

Paths

- A path is a sequence of vertices and edges of a graph – Vertices cannot be repeated. Edges cannot be repeated.
- A path of length k is a sequence of vertices $(v_0, v_1, \ldots, v_k)$, where we have $(v_i, v_{i+1}) \in E$.
- If $v_i \neq v_j$ for all $0 \leq i < j \leq k$ we call the path simple.
- If $v_0 = v_k$ for all $0 \leq i < j \leq k$ and $v_0 = v_k$ the path is a cycle.
- A path from vertex u to vertex v is a path $(v_0, v_1, \ldots, v_k)$ such that $v_0 = u$ and $v_k = v$.

Shortest Paths

- A shortest path between vertices u and v is a path from u to v of minimum length.
- The distance $d(u, v)$ between vertices u and v is the length of a shortest path between u and v.
- If u and v are in different connected component then $d(u, v) = \infty$.

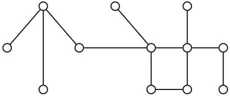


Graph Diameter

- The diameter D of a connected graph is the maximum (over all pairs of vertices in the graph) distance.

$$D = \max_{(u,v):u,v \text{ connected}} d(u, v)$$

- If a graph is disconnected then we define the diameter to be the maximum of the diameters of the connected components.



Breadth-first Search

- Given a graph $G(V, E)$,
- and a distinguished source vertex u,
- breadth-first search systematically explores the edges of G to "discover" every vertex that is reachable from u.
- It computes the distance from u to each reachable vertex.
- It computes a spanning subgraph of G, the "breadth-first tree", with root u that contains all reachable vertices.
- For any vertex v reachable from u, the path in the breadth-first tree from u to v corresponds to a "shortest path" from u to v in G.

Example of Execution of Breadth-First Search Algorithm

Initial Graph

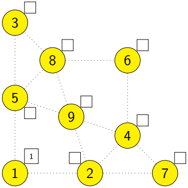
The graph contains 9 vertices, 14 edges

Vertex **1** is the source node.

Vertex **1** is discovered.

Vertices **2,5** are the frontier.

All other vertices are not discovered.



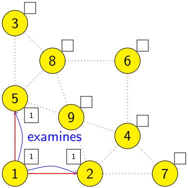
Example of Execution of Breadth-First Search Algorithm

1^{st} Round

Vertex **1** sends examines adjacent vertices.

Vertice **2,5** are discovered.

Vertices **3,4,7,8,9** are the frontier.

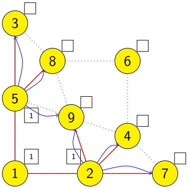


Example of Execution of Breadth-First Search Algorithm

2^{nd} Round

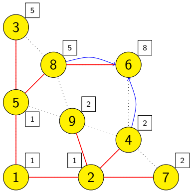
Vertices **3,4,7,8,9** are the discovered.

Vertex **6** is the frontier.



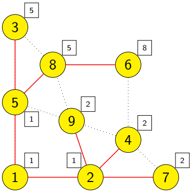
Example of Execution of Breadth-First Search Algorithm

3^{rd} Round
All vertices are discovered.



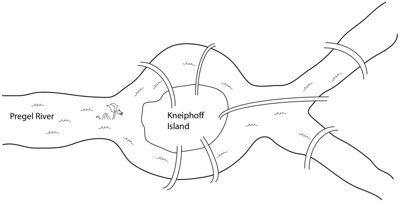
Example of Execution of Breadth-First Search Algorithm

Final Graph
Breadth-first search tree constructed.



Bridges of Königsberg

Euler was interested in whether he could arrange a tour of the city in such a way that the tour visits each bridge exactly once

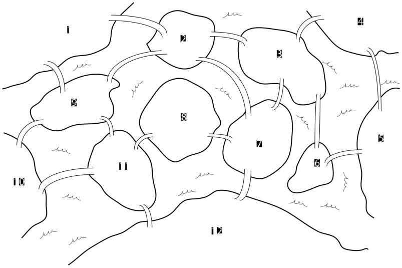


Bridge Problem

Find a tour through a city (located on n islands connected by m bridges) that starts on one of the islands, visits every bridge exactly once, and returns to the originating island.

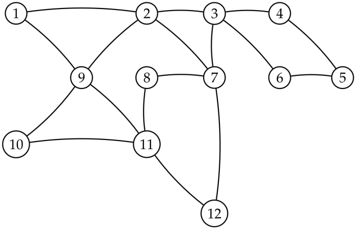
Input: A map of the city with n islands and m bridges.

Output: A tour through the city that visits every bridge exactly once and returns to the starting island.



Transformation of the Map into a Graph

- Every island corresponds to a vertex.
- Every bridge corresponds to an edge.



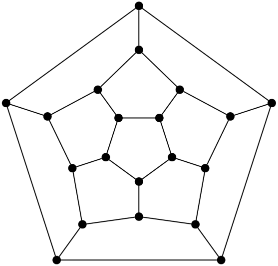
Eulerian Cycle Problem

Find a cycle in a graph that visits every edge exactly once.

Input: A graph G.
Output: A cycle in G that visits every edge exactly once.

Hamilton's Game

- Sir William Hamilton invented a game corresponding to a graph whose twenty vertices were labeled with the names of twenty famous cities.
- The goal is to visit all twenty cities in such a way that every city is visited exactly once before returning back to the city where the tour started.



Hamiltonian Cycle Problem

Find a cycle in a graph that visits every vertex exactly once.

Input: A graph G.

Output: A cycle in G that visits every vertex exactly once.