Graph Theory notes

Abiyaz Chowdhury

January 24, 2019

1 Introduction

Definition 1.1 A graph, is a pair (V, E) where $E \subseteq V \times V$. The set V is called the vertex set (or node set) of the graph, and the set E is called the edge set. The elements of V are the vertices (or nodes) of the graph, and elements of E are the edges. An edge $(u, v) \in E$ is incident on the vertices u and v of the graph. Note that the edge set cannot be empty if the vertex set is empty. In an undirected graph, $(u, v) \in E \implies (v, u) \in E$ whenever $u \neq v$. A directed graph is one that is not undirected.

Definition 1.2 A simple graph $G = (V, E)$ is a graph with the property that for all $v \in V$ we have that $(v, v) \notin E$.

Definition 1.3 A subgraph $G' = (V', E')$ of a graph G is a graph with the property that $V' \subseteq V$ and $E' \subseteq E$. Note that every graph is a subgraph of itself.

Definition 1.4 A weighted graph is a pair (G, f) where G is a graph and $f : E \to \mathbb{R}$ is a mapping that defines the weight of each edge of the graph G. An undirected weighted graph must satisfy the restriction that $f(u, v) = f(v, u)$ for all edges $(u, v) \in E$.

Definition 1.5 In a graph, a neighbor of a vertex u is any vertex v such that $(u, v) \in E$. In an undirected graph, if u is a neighbor of v, then v is a neighbor of u.

Definition 1.6 A path from vertex v_1 to vertex v_n exists in a graph if there exists a sequence of vertices $v_1, ..., v_n$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \{1, 2, ..., n-1\}$. The path is called simple if the vertices $v_1, ..., v_n$ are all distinct with the one exception of the possibility that $v_1 = v_n$. A trivial path is a path of length 0. A nontrivial path is a path of length at least 1. Every vertex in any graph has a trivial path to itself. If a graph has an edge (a, a) for some vertex $a \in V$, then this vertex has a path of length 1 to itself.

Definition 1.7 A cycle is nontrivial path from a vertex to itself. A simple cycle is a nontrivial simpe path from a vertex to itself.

Theorem 1.1 ■ Every path in a graph may be decomposed into a sequence of paths consisting of simple paths and simple cycles.

Definition 1.8 A vertex $u \in V$ in a graph is connected to another vertex $v \in V$ if there exists a path from u to v in the graph.

Theorem 1.2 ■ Every vertex in any graph is connected to itself. In an undirected graph, if $u \in V$ is connected to $v \in V$, then v is also connected to u. For any graph, if u is connected to v and v is connected to w, then u is connected to w.

Theorem 1.3 On a graph $G = (V, E)$, define the relation on uRv on V as u is connected to v. Then for undirected graphs, this relation is an equivalence relation. This equivalence relation partitions V into various equivalence classes. In particular, if $V' \subseteq V$ is one of the equivalence classes induced by this relation, then the subgraph $G' = (V', E')$ where E' contains all edges $(u, v) \in E$ such that $u, v \in V'$ is called a component of the graph G.

Definition 1.9 A connected graph is a graph in which every pair of vertices are connected to one another. The components of a graph are therefore each connected subgraphs of the original graph G.

Definition 1.10 A cyclic graph is a graph that contains a cycle. An acyclic graph is a graph that is not cyclic.

Definition 1.11 A tree is an undirected graph that is both connected and acyclic.

Theorem 1.4 A directed graph is a tree if and only if there is a unique path from any vertex u to any other vertex v in the graph.

Definition 1.12 A leaf is a vertex having only one neighbor.

Theorem 1.5 Every tree has at least two leaves.

Definition 1.13 In a connected graph, a bridge is an edge whose removal would disconnect the graph (cause the graph to no longer be connected).

Theorem 1.6 A directed graph is a tree if and only if every edge of the graph is a bridge.

2 Graph storage and search

Note: All analyses of time complexity assume the RAM model of computation, in which each memory access can be done in constant time. This is a good assumption when the memory addresses fit in a machine word. As of the time of this writing (2019), most modern operating systems are based on a 64-bit architecture, and therefore up to 2^{64} addresses can be supported, which is more than enough for even the largest graphs.

Definition 2.1 A sparse graph is a graph

Definition 2.2 In memory, a graph may be represented in either adjacency matrix or adjacency list format. In the adjacency matrix format, the graph is encoded as a $V \times V$ binary matrix M where V is the size of the vertex set, and where $M[u][v] = 1$ if and only if $(u, v) \in E$. In the adjacency list format, the graph is encoded as an array of linked lists. Each vertex $v \in V$ corresponds to a linked list which lists the neighbors of that vertex v. The adjacency list format is preferred for sparse graphs, and the matrix format is preferred for dense graphs.

Theorem 2.1 The adjacency matrix of any directed graph is symmetric.

Definition 2.3 A graph search is an algorithm that starts at some vertex v and proceeds to visit the graph with the property that every vertex reachable from v, (i.e. having a path from v) will be visited at some point in the search.

Definition 2.4 Any-first search is a graph search that starts with an empty bag, and puts the starting vertex v into the bag. In each iteration, a random vertex is taken from the bag and marked. Its unmarked neighbors are then put into the bag. The algorithm continues until the bag is empty. By the end of the algorithm, the visited vertices are the vertices that are marked by the algorithm.

Theorem 2.2 Any-first search is guaranteed to terminate on any finite graph. In addition, every vertex reachable from the starting vertex will eventually be marked by any-first search. Vertices not reachable from the starting vertex will not be marked. Any-first search is therefore a proper graph search algorithm.