The little book of graph notes

February 22, 2020

Contents

1 On Graphs **5**
1.1 What is a Graph? 5
1.2 Why do we care about Graphs? 5
1.3 On Different Types of Graphs 6
1.3.1 Multigraphs 6
1.3.2 Digraphs 6
1.4 On Vertices and Edges 7
1.5 On Isomorphisms 7
1.5.1 The Handshaking Lemma 8
1.6 On Special Kinds of Graphs 9
1.7 On the Storage of Graphs 9

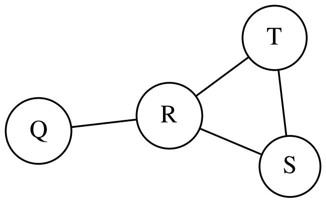
2 Forest for the Trees **11**
2.1 Definitions 11
2.2 Trailblazing 13

Chapter 1

On Graphs

1.1 What is a Graph?

GRAPHS, in their elementary form, are a pair of sets: V (**Vertices**) and E (**Edges**) such that $V \neq \emptyset$ (V cannot be empty), and $\forall e \in E, e = \{v_i, v_j\}, v_i, v_j \in V$ (every edge in E is a pairwise subset of vertices v_i and v_j in V). Here is an example of a very simple graph:



In this graph, our vertex set is $V = \{Q, R, S, T\}$, and our edge set is $E = \big\{\{Q, R\}.\{R, S\}, \{R, T\}, \{S, T\}\big\}$.

Other names for Vertices are **nodes** or **points**. Other names for edges are **links** or **lines**. When we want to refer to the vertex set of a specific graph G, we will denote it as $V(G)$. Similarly, we will denote the edge set as $E(G)$.

1.2 Why do we care about Graphs?

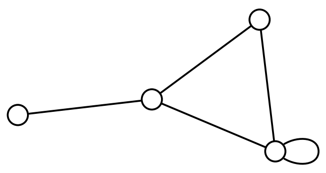
Graphs can be used to model many real world problems, or theoretical ideas. For example, graphs can be used to represent relationships between different objects or groups. They are useful for visualizing data, such as social media connections, or network topology. Additionally, a more practical use is for mapping routes from one location to another, much like the GPS on your phone.

As for more theoretical uses, graphs can represent other structural relationships. Tree structures are a type of graph, as are flow diagrams or DFAs/NFAs from Computation Theory. Perhaps one of their more important uses is in determining where problems fit in P vs. NP.

1.3 On Different Types of Graphs

1.3.1 Multigraphs

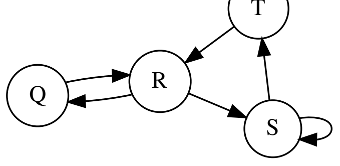
A **multigraph** is a pair of sets (V, E) such that $V \neq \emptyset$, and E is a finite family of unordered pairs of V. That is to say, edges are not necessarily unique subsets, and do not have order. Thus, multigraphs allow for loops from one vertex to itself. Here is an example of a multigraph:



Because multigraphs are unordered, then an edge (a, b) would be the same as (b, a). But what if we only want an edge in one direction but not the other?

1.3.2 Digraphs

A **digraph** is a pair of sets (V, E) such that $V \neq \emptyset$, and E is a set of ordered pairs of elements in V. Informally, we say that a digraph is a directed graph, where we annotate each edge with an arrow to mark the direction it flows. Here is an example of a digraph:



1.4 On Vertices and Edges

There is a lot to be said of the mathematics of graphs, but before we can establish such mathematics, we must first establish some essential definitions. Let G be a graph with vertex set V and edge set E. Let $u, v, w \in V$ be vertices of G.

Join: If $\{u, v\} \in E$, then u is said to **join** v.

Adjacent: u and v are said to be **adjacent** if $\{u, v\} \in E$. Similarly, if $\{u, v\}, \{u, w\} \in E$, then $\{u, v\}$ and $\{u, w\}$ are said to be **adjacent**.

Incident: If $\{u, v\} \in E$, then u and v are said to be **incident** to $\{u, v\}$. Similarly, if $\{u, v\}, \{u, w\} \in E$, then $\{u, v\}$ and $\{u, w\}$ are said to be **incident** to u.

Degree: The **degree** of a vertex, denoted $\deg(v)$ or $\rho(v)$, is the number of edges incident to v.

Isolated: If $\deg(v) = 0$, then we say v is an **isolated** vertex.

Pendant: If $\deg(v) = 1$, then we say v is a **pendant** vertex.

These are a lot of formal definitions, but they will be useful when classifying different graphs and discussing their structures and properties. Informally, we say two edges are adjacent if they share a vertex, and two vertices are adjacent if they share an edge. Similarly, we say that two edges are incident to a vertex if they share that vertex, and two vertices are incident to an edge if they share that edge.

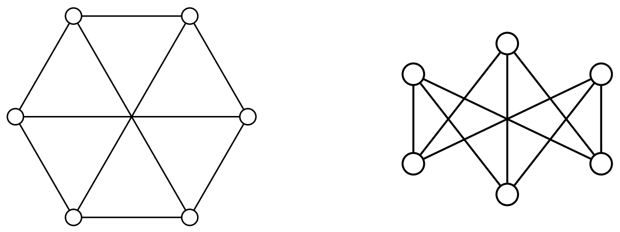
1.5 On Isomorphisms

The term *isomorphic* has its roots in Greek, *iso-* meaning same, and *morphic* referring to form or shape. We say that two graphs are **isomorphic** if they have this "same shape" property, but what does that mean?

Mathematically, we say that the graph G_1 is **isomorphic** to the graph G_2, denoted $G_1 \sim G_2$, if there is a one-to-one correspondence between the vertices V_1 of G_1 and the vertices V_2 of G_2 that preserves adjacencies. This is a mapping f from G_1 to G_2 defined as follows:

$$f : G_1 \mapsto G_2 \text{ such that } \forall \{v_i, v_j\} \in E(G_1), \{f(v_i), f(v_j)\} \in E(G_2)$$

This may look like a lot of intimidating gibberish, but it represents the idea of preserving the edges between two vertices in G_1 after mapping them to G_2. For example, the following graphs are isomorphic:



Isomorphisms are a mapping, and thus have certain propoerties that allow us to classify them with other mappings. Let G_1, G_2, and G_3 be graphs. Isomorphisms are what we call an equivalence relation, as the following three propoerties hold:

Reflexive: $G_1 \sim G_1$

Symmetric: If $G_1 \sim G_2$, then $G_2 \sim G_1$

Transitive: If $G_1 \sim G_2$ and $G_2 \sim G_3$, then $G_1 \sim G_3$

1.5.1 The Handshaking Lemma

We are now armed with enough information to decipher some information hidden in the graphs we deal with. The first of these is called the Handshaking Lemma. Let's tackle this proof:

Lemma 1.1 (The Handshaking Lemma)**.** *Let G be a graph. Then the number of vertices with odd degree is even.*

Proof. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertices of G. The the sum of the degrees,

$$\sum_{i=1}^{n} \deg(v_i) = 2m, m \in \mathbb{Z},$$

is even, since each edge is counted twice (convince yourself that this must be true, if you don't understand). Now, if we were to remove all vertices of even degree, then we would be removing an even number of edges from this sum. This leaves us with

$$\sum_{odd\ degrees}^{n} \deg(v_i) = 2k, k \in \mathbb{Z},$$

which is even. Now, the only way for the sum of odd numbers to be even is for there to be an even number of those odds. Thus, we must have an even number of odd degree vertices in G. □

1.6 On Special Kinds of Graphs

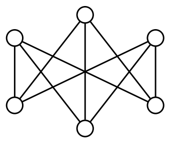
We've talked a lot about the different aspects of graphs, but now let's use this information to actually classify different kinds of graphs, starting with the **null graph**. the null graph is a graph with n vertices but no edges, and is denoted as N_n.

A **regular graph** is a graph G with n vertices whose vertices all have the same degree, and we say such a graph is regular of degree $r < n$.

A **complete graph** is a graph G with n vertices that is regular of degree $n-1$. We denote such a graph by K_n. The total number of edges in the complete graph K_n is given by summing the number of edges:

$$\sum_{i=1}^{n-1} i = \frac{(n-1)(n-2)}{2}$$

A **bipartite graph** is a graph with vertex set $V = V_1 \cup V_2$ where $V_1 \cap V_2 = \emptyset$ and $\forall v_i, v_j \in V_1$ and $v_m, v_n \in V_2$, $(v_i, v_j), (v_m, v_n) \notin E$. This is another way to say that the vertices of a bipartite graph G can be divided into two sets there all edges in G only go between both sets - neither set of vertices has two vertices which make an edge. A complete bipartite graph is denoted $K_{m,n}$, with two sets containing m and n vertices, with all vertices in one set connecting to all the vertices of the other. Here is an example of a bipartite graph, $K_{3,3}$:

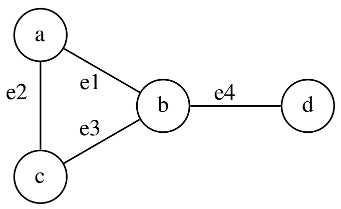


A **subgraph** of $G = (V, E)$ has the vertex set H and the edge set F, where $H \subseteq V$ and $F \subseteq E$, but only the edges of E which correspond to the vertices in H can be in F.

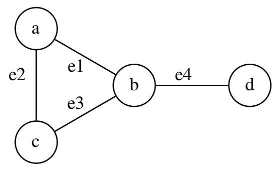
1.7 On the Storage of Graphs

We've talked a lot about graphs and their theories and applications, but how exactly do we represent graphs in a computer? The idea that a graph defines a set of relations in important, so the natural way to consider this is some sort of data structure with nodes which point to their adjacent neighbors. This certainly is an intuitive way to consider this, but examining them mathematically will make our lives easier in the future.

Consider the following graph:



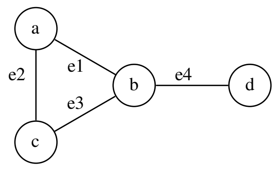
We can store this in what's called an **Adjacency Matrix**, a matrix where each row and column represents a particular vertex in the graph, and entries in the matrix signify that two vertices share an edge. The adjacency matrix for the given graph is:



$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 0 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 1 & 0 & 0 \\ d & 0 & 1 & 0 & 0 \end{array}$$

We usually denote an adjacency matrix $[A_{i,j}]$, where a 1 in entry $a_{i,j}$ identifies that vertices i and j share an edge, and a 0 means that they don't. While adjacency matrices for a graph with n vertices occupies n^2 space, they do provide constant lookup time to see if two vertices share an edge.

We also have an **Incidence Matrix**, which tells us if an edge and a vertex are incident. The incidence matrix for the graph above is:



$$\begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline a & 1 & 1 & 0 & 0 \\ b & 1 & 0 & 1 & 1 \\ c & 0 & 1 & 1 & 0 \\ d & 0 & 0 & 0 & 1 \end{array}$$

It might be preferable to use the adjacency matrix for the symmetry along the main diagonal of the matrix.

Chapter 2

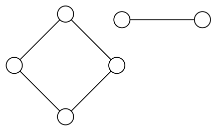
Forest for the Trees

TREES are another special kind of graph which we did not discuss in the previous chapter. Trees are somewhat unique compared to the other types of specialty graphs described before as they allow us certain algorithms to perform to solve different problems. Before we go in depth into trees, we must first define an operation and some other terms.

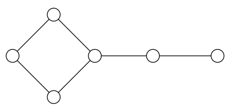
2.1 Definitions

Let G_1 and G_2 be graphs. Then $G_1 \cup G_2$ is the **union** of G_1 and G_2 with the vertex set $V = V(G_1) \cup V(G_2)$ and edge set $E = E(G_1) \cup E(G_2)$. Now, let's establish some more definitions:

Disconnected: A graph is **disconnected** if it is the union of two graphs. Here is an example of a disconnected graph:

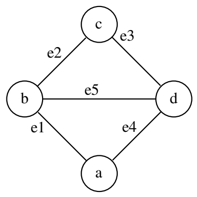


Connected: A graph is **connected** if it is not disconnected (wow). Here is an example of a connected graph:

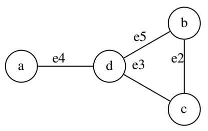


Component: A **component** graph is one graph which makes up part of a union. In $G = G_1 \cup G_2$, G_1 and G_2 are components.

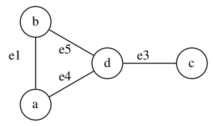
We can also define the **subtraction** of a set of edges F from a graph G. We say that $G - F$ is the graph resulting in the removal of the edges in F from G. Suppose we have the following graph G:



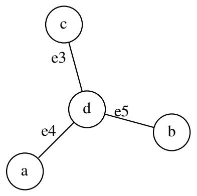
Then the following are different subtractions from G.



(a) $G - \{e1\}$

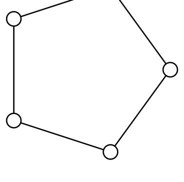


(b) $G - \{e2\}$



(c) $G - \{e1, e2\}$

Let's define a new type of graph. A connected graph that is regular of degree 2 is called a **cycle**, which we denote C_n. Here is an example of a cycle, C_5:



A useful operation on graphs is to take the complement. The **complement** of a graph G is the graph $\overline{G}$, with vertex set $V(\overline{G}) = V(G)$, with two vertices adjacent in $\overline{G}$ if and only if they are *not* adjacent in G.

2.2 Trailblazing

We've defined what graphs are as a construct, but how do we actually navigate from one point to another? There are different ways to navigate from vertex to vertex, as well as different forms of these navigations.