Graph Theory

Author: Isaac FEI

Preface

I mainly refer to (Bondy and Murty 1976).

Contents

Chapter 1 Basic Concepts of Graphs

1.1 Isomorphism

Two graphs G and H are identical, written as $G = H$, if all their components are the same, that is, $V(G) = V(H)$, $E(G) = E(H)$ and $\psi_G = \psi_H$. Identical graphs of course share the same properties. However, a graph H does not necessarily have to be exactly G to preserve all its properties. The labels of the vertices and edges are immaterial.

Definition 1.1.1

Two graphs G and H are said to be isomorphic, written as $G \cong H$, if there exist bijections $\theta : V(G) \to V(H)$ and $\phi : E(G) \to E(H)$ such that

$$\psi_G(e) = uv \implies \psi_H(\phi(e)) = \theta(u)\theta(v) \tag{1.1}$$

*The ordered pair (θ, ϕ) is called an **isomorphism** between G and H.* ♣

(Bondy and Murty 1976) includes the reverse direction of (1.1) in the definition, that is,

$$\psi_G(e) = uv \iff \psi_H(\phi(e)) = \theta(u)\theta(v)$$

But the reverse direction is redundant. To see this, we suppose that $\psi_H(\phi(e)) = \theta(u)\theta(v)$ and $\psi_G(e) = xy$. By (1.1), we have $\psi_H(e) = \theta(x)\theta(y)$. It then follows that $\theta(u)\theta(v) = \theta(x)\theta(y)$. We have either $\theta(u) = \theta(x)$, $\theta(v) = \theta(y)$, or $\theta(u)\theta(y)$, $\theta(v) = \theta(x)$. Because θ is a bijection, either $u = x$, $v = y$, or $u = y$, $v = x$. Either way, we have $uv = xy$. Therefore, $\psi_G(e) = xy = uv$, which proves the reverse direction $\Leftarrow$.

For simple graphs, there is no need to find a bijection between edges once the bijection θ between vertices is established.

Proposition 1.1.1

Let G and H be simple graphs. Then $G \cong H$ if and only if there exists a bijection $\theta : V(G) \to V(H)$ such that

$$uv \in E(G) \implies \theta(u)\theta(v) \in E(H) \tag{1.2}$$

♠

Proof (Necessity) Suppose that there exist θ and ϕ satisfying (1.1). If $e = uv \in E(G)$, then by (1.1), $\psi_H(\phi(e)) = \theta(u)\theta(v)$, which implies $\theta(u)\theta(v) \in E(H)$.

(Sufficiency) Define $\phi : E(G) \to E(H)$ by

$$\phi(uv) = \theta(u)\theta(v)$$

We need to show ϕ is bijective. Suppose $\phi(uv) = \phi(xy)$. We have $\theta(u)\theta(v) = \theta(x)\theta(y)$. Applying a similar argument we used in the previous comments, we will finally obtain $uv = xy$, which means ϕ is injective. On the other hand, for any edge $f \in H$. Write $f = ij$ (i.e., $\psi_H(f) = ij$). Then because θ is bijective, there exist $u, v \in V(G)$ such that $\theta(u) = i$ and $\theta(v) = j$. Hence, $\phi(uv) = ij$, which implies ϕ is surjective.

If $\psi(e) = uv$, i.e., $e = uv \in E(G)$, then we have $\theta(u)\theta(v) \in E(H)$ by (1.2). Equivalently, $\psi_H(\phi(e)) = \theta(u)\theta(v)$. ■

A **complete bipartite graph** is a *simple* bipartite graph with bipartition (X, Y) in which each vertex in X is incident with each vertex in Y. That is, if $x \in X$ and $y \in Y$, then $xy \in E$. If $|X| = m$ and $|Y| = n$, we often use the symbol $K_{m,n}$ to denote this complete bipartite graph. (See Figure 1.1.) Note that this implicitly implies that the complete bipartite graph is unique in some way since we can represent it with a common symbol. Indeed, it is unique up to isomorphism, as we will show in the next proposition.

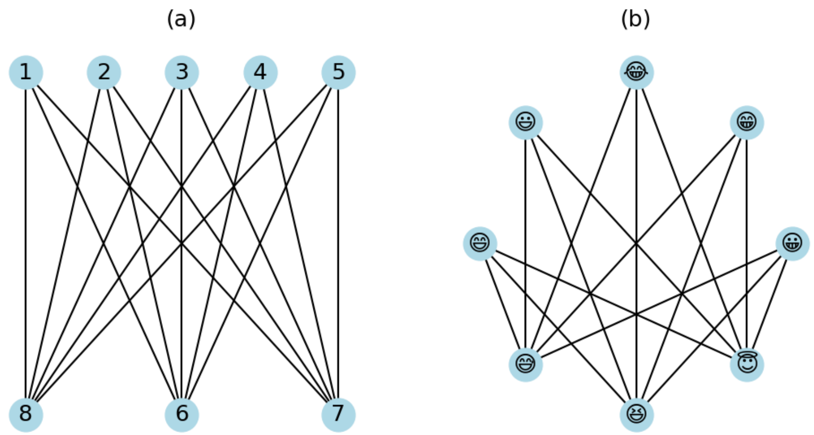

Figure 1.1: Both (a) and (b) are $K_{5,3}$.

Proposition 1.1.2

Let $G[X, Y]$ and $H[U, V]$ be two complete bipartite graphs with $|X| = |U|$ and $|Y| = |V|$. Then $G \cong H$. In other words, a complete bipartite graph is unique up to isomorphism if the sizes of its two vertex sets in bipartition are determined.

Proof Since $|X| = |U|$ and $|Y| = |V|$, we can find a bijection $\theta : V(G) \to V(H)$ in such a way that θ maps each point in X onto U, and each point in Y onto V. Then for an edge $xy \in E(G)$, we have $\theta(x)\theta(y) \in E(H)$ since there has to be an edge connecting $\theta(x) \in U$ and $\theta(y) \in V$ by the definition of complete bipartite graphs. This proves $G \cong H$ by Proposition 1.1.1. ■

1.2 Vertex Degrees

1.3 Paths and Connection

Proposition 1.3.1

If there is a (u, v)-walk in G, then there is also a (u, v)-path in G.

This can be proved easily using the following algorithm (Algorithm 1).

Algorithm 1: Extracting a Path From a Walk

Input: A walk $W = v_0 e_1 v_1 \cdots e_k v_k$

Output: A path P

initialize P as a sequence containing just one vertex v_0 ;
for $i = 1, \dots, k$ **do**
  if *v_i is not in P* **then**
    append e_i and v_i to P ;
  else
    remove all the vertices and edges after the vertex v_i from P ;
  end
end

Proposition 1.3.2

The number (v_i, v_j)-walks of length k in G is the (i, j)-th entry of the k-th power of the adjacency matrix A, i.e., A^k.

Proof ∎

1.4 Cycles

One simple yet useful observation of a particular longest path in a graph is that all the neighbors of the terminus must occur along the path. To be specific, if $P = v_0 e_1 v_1 \cdots e_k v_k$ is one of the longest paths in G then P must contain all vertices in $N(v_k)$. To prove this, we assume P does not contain $v_{k+1} \in N(v_k)$. (Suppose $\psi(e_{k+1}) = v_k v_{k+1}$.) Then the path $P + e_{k+1} v_{k+1}$ is clearly longer than P, which leads to a contradiction. Figure 1.2 depicts such an example. Note that if 8 were a neighbor of 7, then path 12345678 would be longer.

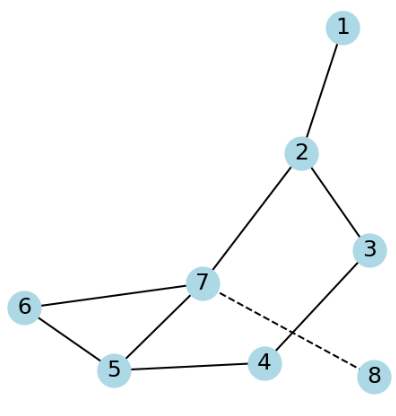

Figure 1.2: Path 1234567 is one of the longest paths.

Proposition 1.4.1

If $\delta(G) \geq 2$, then we can find a cycle starting from each vertex of G.

Proof The conclusion is trivial if G has a loop. Now, we assume that G contains no loops.

Note *Note that there are cases where G has no paths if we allow it to have loops. For example, if every vertex of G is incident with just exactly one loop, then G still satisfies the hypothesis. But there are no paths in G.*

Let $P = v_0e_1v_1 \cdots e_{k-1}v_{k-1}e_kv_k$ be one of the longest paths in G. Since $\deg(v_k) \geq 2$ and v_k has no loops, v_k has a neighbor, say u, other than v_{k-1}. As noted before, u must occur in P. Therefore, there exists a cycle from u to u. ■

In fact, we have an algorithm to find a cycle without knowing the longest path in G.

Algorithm 2: Finding a Cycle in G With $\delta(G) \geq 2$

Input: G with $\delta(G) \geq 2$
Output: A cycle C

```
1  if G has a loop e from v to v then
2      C ← vev ;
3      return C ;
4  end
5  pick a vertex v_0 ;
6  P ← v_0 ;
7  pick v ∈ N(v_0) and let e be the corresponding edge, i.e., ψ(e) = v_0 v ;
8  while v is not in P do
9      P ← P + ev ;
10     pick u ∈ N(v) such that there exists an edge f satisfying ψ(f) = vu and f ≠ e ;
11     v ← u ;
12     e ← f ;
13 end
14 remove from P all vertices and edges before v ;
15 C ← P + ev ;
```

Proof We need to show that Algorithm 2 works correctly.

(initialization) Firstly, note that line 7 is possible since v_0 has no loops and $\deg(v_0) \geq 2$.

We claim the loop invariants are

1. P has j vertices assuming that we are to execute the j-th iteration,
2. P has no duplicated vertices, i.e., P is a path, and
3. edge e is incident with v.

(Maintenance) Suppose we are in the j-th iteration. After line 9, P remains a path. Because $\deg(v) \geq 2$, there exists an edge f other than e that is incident with v. Hence, line 10 works correctly. After executing line 12, we find that the number of vertices in P is increased by one, i.e., $j+1$, P is still a path and e is incident with v.

(Termination) We can complete at most $n-1$ iterations since P can hold at most as many vertices as there are in G. Upon termination, we find v is in P and e is incident with v. By removing from P all vertices and edges before v and then append to it edge e and vertex v, we will obtain a cycle from v to v. ■

References

[1] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. New York: North Holland, 1976. ISBN: 978-0-444-19451-0.

Index