CHAPTER 6
Parallel Graph Algorithms

One area in which a great deal of work has been done in the development of parallel algorithms is that of graph algorithms. An undirected graph \( G \) is a pair of \((V,E)\), where \( V \) is a finite set of points called vertices and \( E \) is a finite set of arcs called edges. An edge \( e \in E \) is an unordered pair \((v,u)\) where \( v,u \in V \) and vertices \( v \) and \( u \) are connected. Similarly, a directed graph \( G \) is a pair of \((V,E)\) where \( V \) is a finite set of points called vertices and \( e = (v,u) \in E \) is an ordered pair of vertices, meaning that there is a connection from \( v \) to \( u \). Throughout this chapter the term graph refer to both directed and undirected graphs. Figure 6.1 illustrates a directed and an undirected graph. Many definitions are common to directed and undirected graphs, and in the following some of these definitions are presented. In addition, there are many introductory texts, for example [Sahni 85], which may be consulted.

Definitions

- If \((v,u)\) is an edge in an undirected graph, then \((v,u)\) is incident on vertices \( v \) and \( u \).
- If \((v,u)\) is an edge in a directed graph, then \((v,u)\) is incident from vertex \( v \) and incident to vertex \( u \).

For example, in Figure 6.1 (a), edge \( e_1 \) is incident from vertex 1 and incident to vertex 4, but in Figure 6.1 (b), edge \( e_3 \) is incident on vertices 1 and 4.

- If \((v,u) \in E\) is an edge in an undirected graph \( G \), then \( v \) and \( u \) are said to be adjacent, and if the graph is directed, then vertex \( v \) is said to be adjacent to vertex \( u \).
- A path in graph \( G \) from vertex \( v_1 \) can be defined as a sequence of vertices \( \{v_1,v_2,...,v_k\} \) such that \((v_i, v_{i+1}) \in E\) for all \( 1 \leq i \leq k \).
- A cycle is a path in which the start vertex and the end vertex are the same, and a graph with no cycle is called an acyclic graph.
- A simple cycle in a graph is a cycle in which no vertex occurs more than once in the cycle.

For example, in Figure 6.1 (a) there is a path from vertex 1 to vertex 4 shown as \( \{1,3,5,4\} \), and there is a directed simple cycle shown as \( \{2,5,4,3,2\} \).
• An undirected graph $G$ is connected if every pair of vertices is connected by a path.
• A graph $G' = (V', E')$ is a subgraph of graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.
• A complete graph is a graph such that each pair of vertices is adjacent.
• A tree is a connected acyclic graph in which $|E| = |V| - 1$, and a forest is an acyclic graph.

![Figure 6.1. (a) A directed graph. (b) An undirected graph.](image)

The following statements about a graph $G$ are equivalent:

• $G$ is a tree.
• $G$ is connected with $n$ vertices and $n - 1$ edges.
• $G$ has $n$ vertices, $n - 1$ edges, and no cycles.
• $G$ is such that each pair of vertices is connected by a unique chain.

Sometimes a weight can be associated with each edge in $E$ of the graph $G = (V, E)$. These are real numbers representing the measure of performance of that associated edge, i.e., the cost or benefit of traversing that edge.

• A graph with the associated weight of each edge is called a weighted graph and can be represented as $G = (V, E, W)$, where $W: E \Rightarrow R$ is a real-valued function defined on $E$, and $V$ and $E$ are defined as before. The weight of the graph is defined as the total of the weights of its edges. Similarly, the weight of a path is the sum of the weights of its edges.

Representing a graph in an algorithm may be accomplished via two different methods: adjacency matrix and linked lists.

• Consider a graph with $n$ vertices that are numbered $1, 2, \ldots, n$. The adjacency matrix of this graph can be defined as an $n \times n$ matrix $A = (a_{ij})$ with the following properties: