**Written Assignment Unit 3**

University of the People

CS 3304 Analysis of Algorithms

Romana Riyaz, Instructor

July 10, 2024

A diagram of a network

Description automatically generated

**As part of your assignment, you must determine and include in your discussion whether the graph is:**

* Acyclic or not
* Directed or undirected
* Connected or not
* Simple or not

**Answer:**

* Acyclic
* Directed
* Connected
* Simple

From the image, we can see it has a root, vertices, nodes, arrows, and connections. Therefore, the image structure is represented as:

V= {1,2,3,4,5,6,7,8} where V is the set of vertices.

E= {(1,2),(1,3),(2,3),(2,4),(3,5),(4,7),(5,4),(5,6),(6,7),(7,8)} where E is the set of edges connecting the vertices.

In this tree structure, we can observe:

1. **Acyclic**: This image is acyclic because it does not have loops or cycles in its structure. According to Hazelcast (2023), acyclic means that there are no loops (i.e., “cycles”) in the graph, so for any given vertex, if you follow an edge that connects that vertex to another, there is no path to get back to the initial vertex. This image meets these conditions, so it is acyclic.
2. **Directed**: The graph is directed. As we can see, the image has arrows indicating direction from one vertex to another, proving it is directed.
3. **Connected**: The graph is connected. Each pair of vertices has a path connecting them, indicating a connected structure.
4. **Simple**: The graph is simple. This means there are no multiple edges between any two vertices. We do not find any multiple edges in this picture.

Overall, this analysis ensures that the graph meets all the specified conditions: it is acyclic, directed, connected, and simple.

**Reference**

Hazelcast. (2023, December 13). *Directed Acyclic Graph (DAG) Overview & Use Cases | Hazelcast*. https://hazelcast.com/glossary/directed-acyclic-graph/

Shaffer, C. A. (2010). *A Practical Introduction to data Structures and algorithm Analysis third edition (C++ Version)* (3rd ed.). https://people.cs.vt.edu/~shaffer/Book/C++3e20100119.pdf