A screenshot of a computer

Description automatically generated

Graph as a picture

A diagram of a number

Description automatically generated

**Graph:**

This diagram provides a clear view of the nodes and their connections, showing the parent-child relationships and the weights of the edges.

A **directed graph (or digraph)** is a type of graph where edges have directions. This means each edge has a start vertex and an end vertex, indicating a one-way relationship.

In a directed graph, you can only travel from the start vertex to the end vertex in the direction of the edge. If there's an edge from s to 0, you can go from s to 0, but not necessarily from 0 to s unless there's another edge explicitly connecting 0 back to s.

This diagram shows a directed graph where you can see the direction from s to t through intermediate nodes 0 and 1, with the respective weights of each edge.

*Vertices:* 4 (s, 0, 1, t)

*Edges:* 3 ((s, 0, 1), (0, 1, 5), (1, t, 0.5))

*Vertices (Nodes):*

* Vertices are the fundamental units of a graph.
* In the context of a graph, each vertex represents a distinct entity or point.
* For example, in a social network graph, each vertex could represent a person.

*Edges:*

* Edges are the connections between vertices in a graph.
* An edge connects two vertices and can represent a relationship or a link between those entities.
* Edges can have weights (representing the cost, distance, or any other metric) associated with them.
* In an undirected graph, edges don't have a direction, meaning the relationship is bidirectional.
* In a directed graph, edges have a direction, indicating a one-way relationship from one vertex to another.

**Nodes:**

Node s:

* Represents the starting or root node of the graph.
* It has one child node: 0.

Node 0:

* Represents an intermediate node.
* It is a child of node s.
* It has one child node: 1.

Node 1:

* Represents another intermediate node.
* It is a child of node 0.
* It has one child node: t.

Node t:

* Represents the terminal or end node.
* It is a child of node 1.

**Edges:**

* Edge from Node s to Node 0 with weight 1.
* Edge from Node 0 to Node 1 with weight 5.
* Edge from Node 1 to Node t with weight 0.5.