**Difference between Trees and Graphs**

|  |  |  |
| --- | --- | --- |
|  | **Trees** | **Graphs** |
| **Path** | Tree is special form of graph i.e. **minimally connected graph** and having only one path between any two vertices. | In graph there can be more than one path i.e. graph can have uni-directional or bi-directional paths (edges) between nodes |
| **Loops** | Tree is a special case of graph having no **loops**, no **circuits** and no self-loops. | Graph can have loops, circuits as well as can have **self-loops**. |
| **Root Node** | In tree there is exactly one root node and every **child** have only one **parent**. | In graph there is no such concept of **root** node. |
| **Parent Child relationship** | In trees, there is parent child relationship so flow can be there with direction top to bottom or vice versa. | In Graph there is no such parent child relationship. |
| **Complexity** | Trees are less complex then graphs as having no cycles, no self-loops and still connected. | Graphs are more complex in compare to trees as it can have cycles, loops etc |
| **Types of Traversal** | Tree traversal is a kind of special case of traversal of graph. Tree is traversed in **Pre-Order**, **In-Order** and **Post-Order** (all three in DFS or in BFS algorithm) | Graph is traversed by **DFS: Depth First Search** and in **BFS : Breadth First Search algorithm** |
| **Connection Rules** | In trees, there are many rules / restrictions for making connections between nodes through edges. | In graphs no such rules/ restrictions are there for connecting the nodes through edges. |
| **DAG** | Trees come in the category of **DAG : Directed Acyclic Graphs** is a kind of directed graph that have no cycles. | Graph can be **Cyclic or Acyclic**. |
| **Different Types** | Different types of trees are : **Binary Tree , Binary Search Tree, AVL tree, Heaps**. | There are mainly two types of Graphs : **Directed and Undirected graphs**. |
| **Applications** | Tree applications : sorting and searching like Tree Traversal & Binary Search. | Graph applications : Coloring of maps, in OR (**PERT & CPM**), algorithms, Graph coloring, job scheduling, etc. |
| **No. of edges** | Tree always has **n-1** edges. | In Graph, no. of edges depend on the graph. |
| **Model** | Tree is a **hierarchical model**. | Graph is a **network model**. |
| **Figure** | [http://freefeast.info/wp-content/uploads/2013/06/tree.jpg](http://freefeast.info/general-it-articles/difference-between-trees-and-graphs-trees-vs-graphs/attachment/tree/) | [http://freefeast.info/wp-content/uploads/2013/06/Graph.jpg](http://freefeast.info/general-it-articles/difference-between-trees-and-graphs-trees-vs-graphs/attachment/graph-2/) |