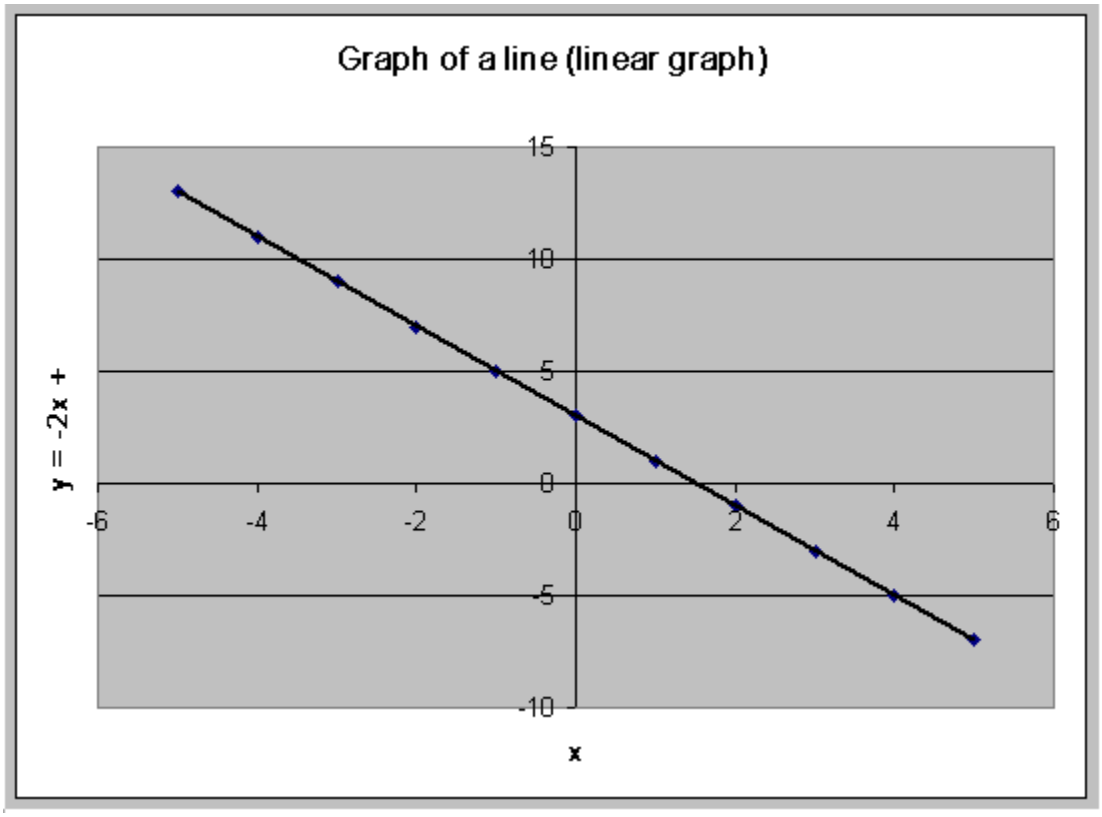
Graphs and Trees

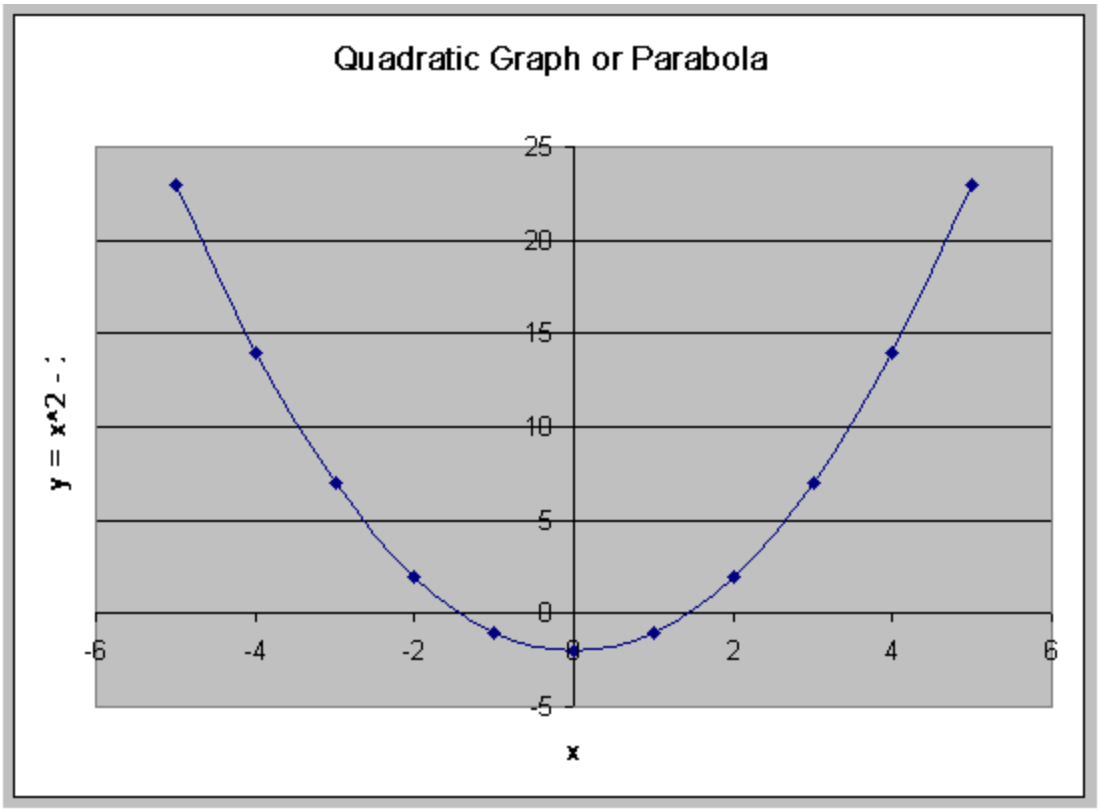
Graphs and Trees

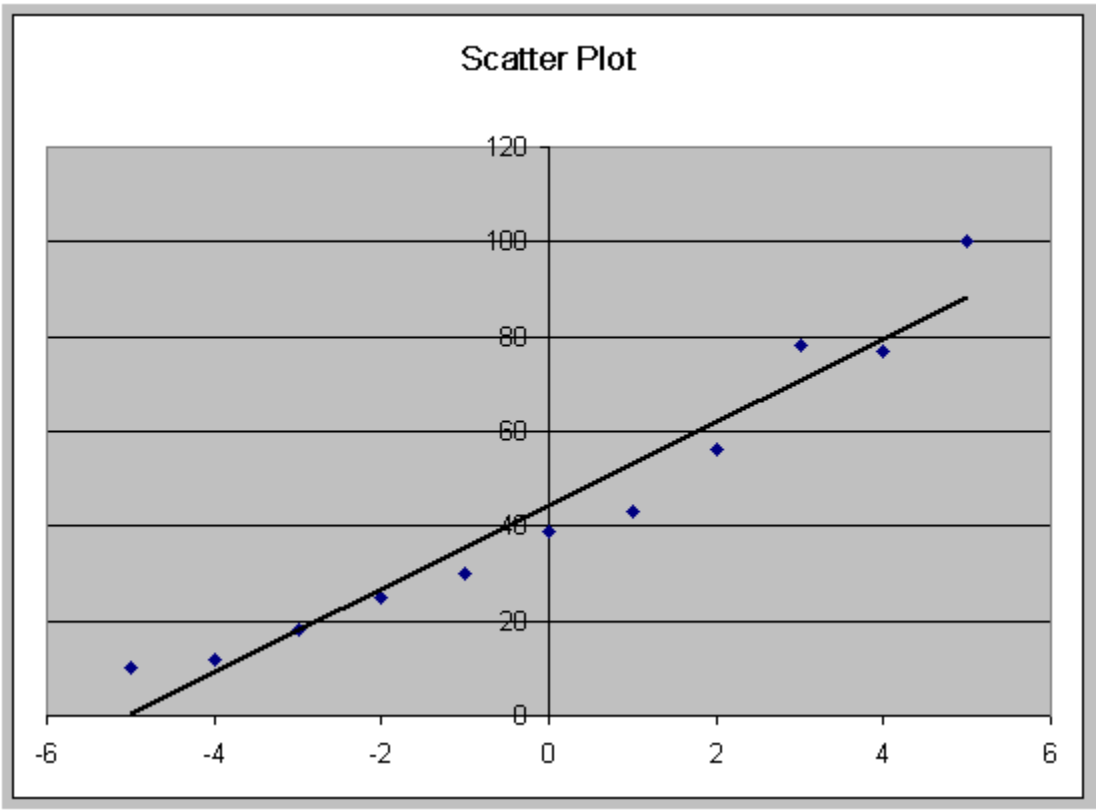
One of the most powerful tools in data analysis, organization, and storage is visualization. From designing the specifications of a database to illustrating the relationships between the contained data types, graphing and related visual tools are an ideal mechanism. While datasets and databases can often contain immense amounts of interrelated data, graphical visualization can offer considerable simplification and can improve both comprehension and subsequent communication.

Understanding Graphs

While the word graph has adopted many meanings, it has a specific meaning in the context of discrete mathematics and mathematics in general. A graph can be thought of as a collection of points that can be connected to each other. This basic definition encompasses many types of familiar graphs, such as the graph of a line; the graph of a quadratic equation, namely a parabola; a scatter plot; or a bar graph.




Quadratic Graph or Parabola
y = x^2 - ;
x
25
20
15
10
5
0
-5
-6
-4
-2
0
2
4
6


Scatter Plot
120
100
80
60
40
20
0
-6
-4
-2
0
2
4
6

However, this definition can also be relaxed and extended into a generalized definition for a mathematical graph. By definition, a graph simply contains vertices and edges. Vertices can be points or can contain textual or numerical meaning. The edges that connect the vertices can be lines, can also be weighted or valued, and can specify direction. For example, data in a database can be contained within the vertices of a graph. Then, these vertices can be connected using edges to show the relationships between the data. If the edges are weighted, this might signify value or cost in the relationship. Maps are a good example of graphs because the cities are the vertices and the weighted edges connecting the cities specify distance and accessibility.

Understanding Trees

A tree, in mathematical terms, is a special type of graph. Often, especially in a database system, hierarchical relationships exist between certain data types. For example, in an educational institution, the administrative hierarchy might include a president at the top, then a vice president, then perhaps administrative assistants, then professors and staff, and then students. While these relationships can be displayed using a basic graph, they can be more accurately illustrated using a tree.

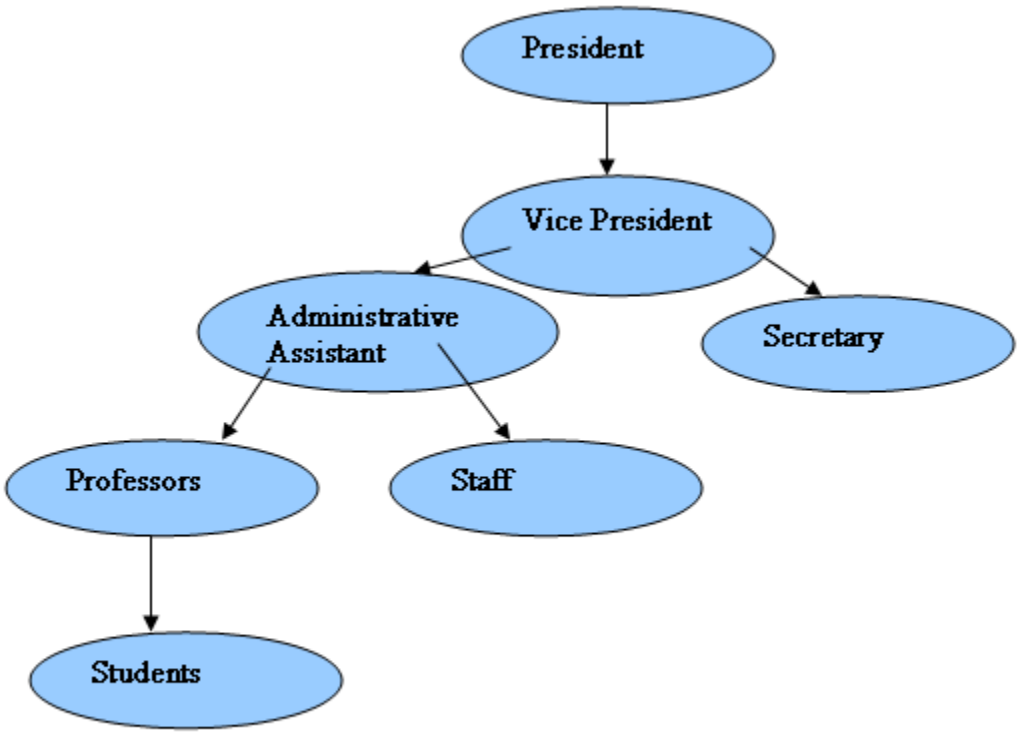


Figure: A hierarchical structure represented by a tree (a tree is a specialized type of graph)

Figure: A hierarchical structure represented by a tree (a tree is a specialized type of graph)

Trees are graphs that have exactly one top vertex, or root. This root would represent the top level of the hierarchy. Next, the root has vertices connected to it known as children. These children are connected to the root with directed edges that always point from the root downward. This notion is continued as the children have children of their own until the tree ends. Trees, like graphs in general, are powerful tools in the visualization of data and numerical relationships.

Traversing, Searching, and Sorting

Both graphs and trees offer further functionality beyond visualization. Once data are placed in either a graph or tree format, data can be searched, sorted, and traversed with significantly greater ease and organization. Within the area of computer science, there are well-known computer algorithms that can traverse or travel through a tree or graph. Similarly, there are algorithms that can search for a data item within a graph or tree or can sort items using a graph or tree. These methods are invaluable when utilizing a database for organizing and analyzing information. For example, once a database is developed and contains related data, a user may want to extract information from the database in an orderly fashion. As such, the user would want to search the database and then display the results as a sorted list. This type of application is best accomplished by storing the data into either trees or graphs and then using *search* and *sort* algorithms to return the desired results.