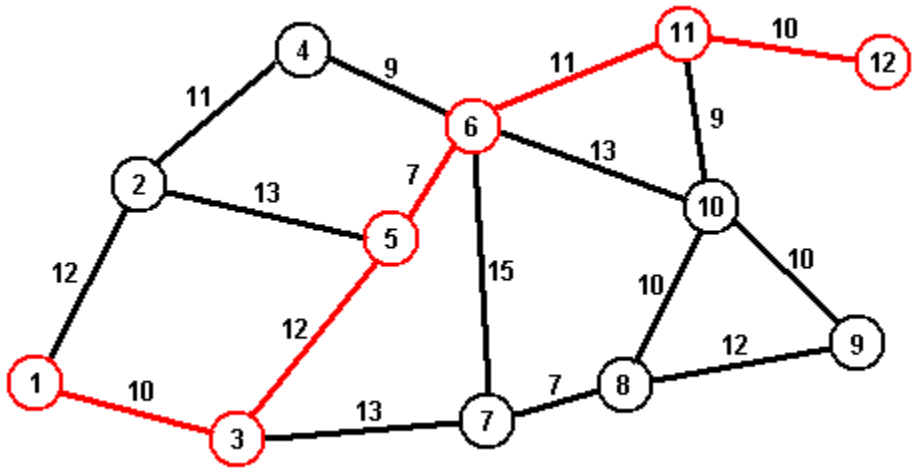
Graph Theory

What is a graph?

A graph is a way to represent a network. So what is a network then? A network is a set of points connected together.



The above diagram is an example of a graph. So the circles are the points and they are numbered from 1 to 12, so every circle has a unique number that allows us to distinguish it from the rest of the points, Formally , the points are called nodes (node) or vertices (vertex). These nodes are connected together with some links. The links are called edges (edge) or arcs (arc).

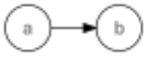
These edges could have a number associated with them, that number is known as the weight.

So node 1 is connected to node 2 with and edge that has a weight 12.

The significance of the weight depends on the problem that we are trying to solve with a graph, for example if the graph represents a city map where the edges are the roads and the nodes are their intersections , then the weight assigned to every edge could represent the distance between the two nodes. But note that the weight could represent anything else for instance it could be negative.

If edges has weights on them then we say that the graph is weighted, otherwise it is un-weighted.

We could represent every edge with the triplet {u,v,w} where u is the first node, v is the second node and w is the weight of the edge. So node u is connected to node v with weight w.

If “u is connected to v” always implies that “v is connected to u” then we can say that the graph is un-directed , otherwise it is directed .  (An extra sharp head to indicate direction)

The graph example shown above is an un-directed graph. The network of rivers is an example of an directed graph because the water flows in only one direction and can't go back.

A path in a graph is a sequence of edges that allows movement from one node to another, for example the edges { {1,3,10} , {3,5,12} , {5,6,7} , {6,11,11} , {11,12,10} } defines a path from node 1 to node 12 , in other words if you start at node 1 and follow those edges in the order they are represented , then you will be able to reach the node 12. The path presented is made in red in the diagram.

You can represent the path in a simpler way by just mentioning the names of the nodes, so the previous path would be expressed like this : {1 , 3 , 5 , 6 , 11 , 12}.

Graph could be represented with three major data structures : (Data structures Chapter)

- Edge list : An array that contains the list of all the edges. This way of representation happens to be very useful in some problems (Minimum Spanning Tree problem …)
- Adjacency matrix : A Boolean square matrix M (2 dimensional array) that indicates that node u and v are connected if M[u][v] has value true , unconnected otherwise. In the case of weighted graphs, M[u][v] would have the weight instead if they are connected and some dummy value otherwise.
- Adjacency List : An array of lists A that indicates the node u is connected to v if the list A[u] contains v. In the case of directed graph , every node is stored alongside its weight, so the edge {u,v,w} would be represented as: A[u] contains the pair {v,w}.

If a path can be written as some : {v1, v2 , v3 … vn , v1} then it is considered a cycle. (Note that we visit every node once). For example the path {1,3,5,2,1} is cycle.

Traversing a graph could be performed with the two fundamental algorithms :

- Depth First Search (DFS)
- Breadth First Search (BFS)

Example graph problems:

- Connected Components
- Shortest Path
- Minimum Spanning tree
- Minimum Cut
- …