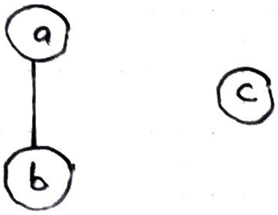
SW10

Graphs:



Vertices (nodes) a, b, c and edge (a, b)
This graph is disconnected

	a	b	c
a	0	1	0
b	1	0	0
c	0	0	0

There are more different types of graphs:
- Complete graph
- directed graph
- tree
- weighted graph
- multigraph

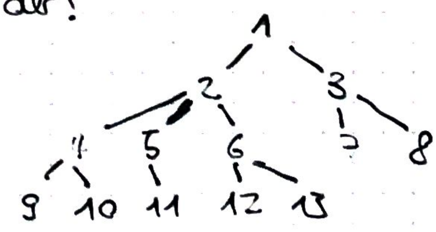
A Graph $G = (V, E)$ consists of a finite V of vertices and a set $E \subseteq V \times V$ of edges.

A weight function on $G = (V, E)$ is a function assigning a real number to each edge $e \in E$.

Depth-First Search (DFS):

1. Start at a, put it on the stack
2. When ever there is an unmarked neighbour, go there and put it on the stack.
3. If there is no unmarked neighbour, return current as result, remove from stack backtrack to the last node on the stack.

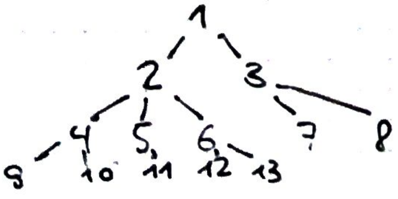
→ Alphabetic order!



Result: 9, 10, 4, 11, 5, ...
Stack: 1, 2, ~~4~~, ~~9~~, ~~10~~, ~~5~~ ...

Breadth-First Search (BFS):

1. Start at a, put it in queue
2. Return first vertex from queue. Mark all neighbours and put them in queue.
3. Do so until queue is empty.



Result: 1 2 3 4
Queue: ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5 6 7 8 9 ...

Spanning trees:

Given a graph G(V, E) with positiv edge weights.

Find a set of edges that connects all vertices of G and has minimum total weight.

Optimistic and Pessimistic approach to find it.

Additional approaches are prim's algorithm and Kruskal's algorithm, (which is the same like the optimistic approach).

All three algorithms are equivalent and find a minimum spanning tree.

Shortes path:

Given a weighted graph G = (V, E), two vertices u, v ∈ V.

Find:

① - Shortest path in G connecting u to v

① - Shortest paths from start vertex V_0 to all other vertices

② - Shortest path between all pairs of vertices.

① Best is Dijkstra's algorithm

② Best is Floyd-Warshall algorithm