Graphs

A graph consists of a set of nodes and a set of edges. Each edge connects 2 nodes.

Basic terms:

directed / undirected: can only go from a to b (or b to a), undirected can go both ways

weighted / unweighted: edge have weights of no weights (exists connection or not)

connected: you can go from any node to any other node (undirected)

trail / path: a set of edges you traverse from a to b

cycle: a path but the beginning and the end is the same

loop: an edge that connects the same node

multi-edge: multiple edges that connects the same 2 nodes

degree: how many edges connected to a node (undirected)]

in-degree: how many edges connected and point inwards

out-degree: how many edges connected and point outwards

star, jellyfish, cactus, bamboo, tree binary tree, complete, bipartite

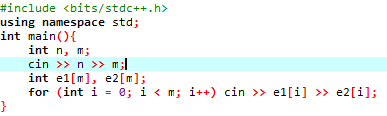
some examples of graph problems:

shortest path

all pairs shortest path

How to represent / store graphs

edge list: a list of edges



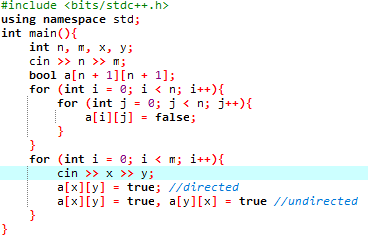
good for enumerating edges

good for sorting edges by weights

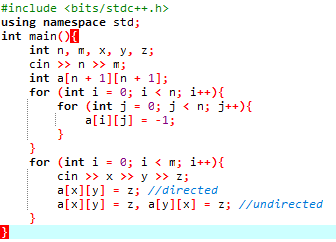
adjacency matrix: a table representing the graph

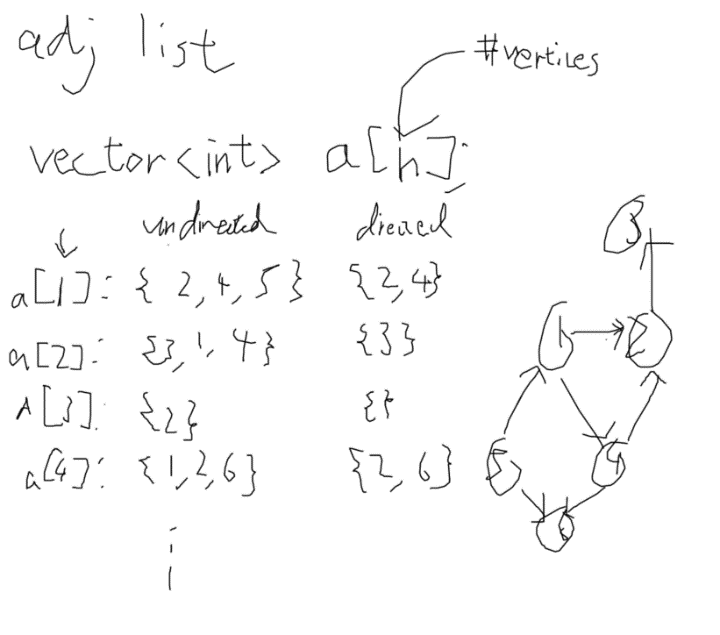
good for easy implementation

unweighted:

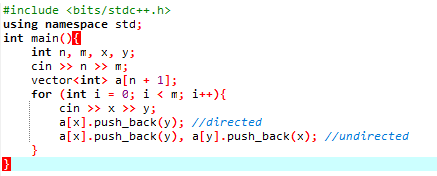


weighted:

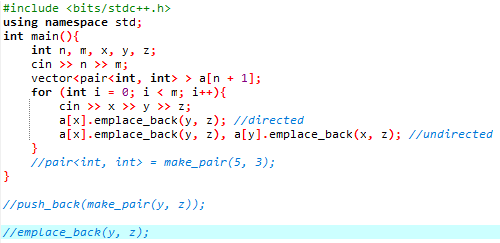


adjacency list: an array of vectors representing the graph

unweighted:



weighted:



good for almost everything

