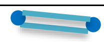
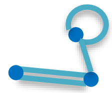
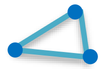
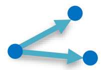
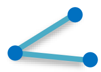
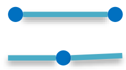
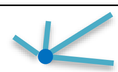
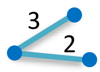
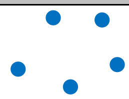
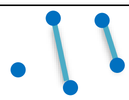
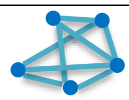
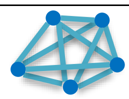
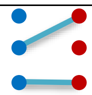
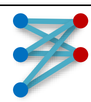
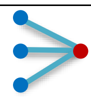
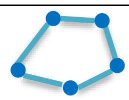
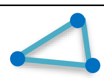
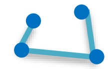
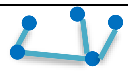
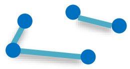
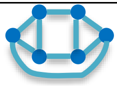
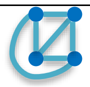
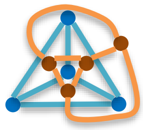
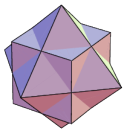
GRAPH THEORY

Introduction

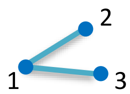
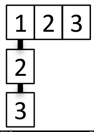
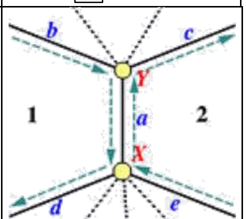
Graph G	A graph is an ordered pair $G = (V, E)$ where V is a finite set of elements and E is a set of 2-subsets of V.		$V = \{1,2,3\}$ $E = \{\{1,2\},\{1,3\}\}$
Edges E	connection between two vertex. $E \subseteq V \times V$		$\|E\| = n_E \geq 0$
Vertices V (nodes)	point where two or more lines meet		$\|V\| = n_V > 0$
Face F	plane between lines/edges		$\|F\| = n_F \geq 0$
Euler	for connected planar subdivisions (we also count the face around)		$2 \leq n_V - n_E + n_F$
Loop	Is an edge which start and ends at the same vertex		
Multiple edge	Is an edge where there has been already an edge		
Multigraph	A multigraph is a graph with multiple edges. Multigraphs are allowed to have multiple edges and loops		
Simple graphs	Graphs without loops, nor multiple edges		
Directed graph	Each edge has a associated direction. The order of two vertices defining an edge matters.		$E = \{(1,2),(1,3)\}$
Order of the graph	number of vertices		$\|V\| = 3$
Size of the graph	number of edges		$\|E\| = 2$
Adjacent **= Neighbors**	either two vertexes which are connected by an edge or two edges which have a common vertex		
Degree of a vertex	number of edges through this vertex = open neighborhood $\deg_G(v) = \|N_G(v)\|$		$\deg_G(v) = 4$
Weighted graph	Each edge is assigned a real number as its "weight". A weight function assigning a real number to each edge $e \in E$.		
Diameter of a graph	The longest distance between two nodes is called the diameter of the graph. In weighted graph.		$dia = 5$

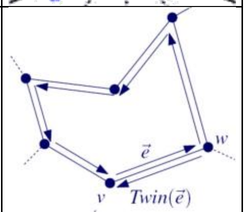
Families of Graphs

Empty graph	No edges between vertexes		$\|V\| = n$ $\|E\| = 0$
Sparse Graph	A graph in which the number of edges is **much less** than the possible number of edges.		$\|E\| \ll \|E_{max}\|$
Dense Graph	A graph in which the number of edges is **close to** the possible number of edges.		$\|E\| \gg \|E_{min}\|$
Complete Graphs K_n	Each vertex connects each other.		$\|V\| = n$ $\|E\| = \frac{n * (n-1)}{2}$
Bipartite Graph	A graph whose vertex set can be partitioned into 2 sets V_1, and V_2 such that every edge $uv \in E$ has $u \in V_1$ and $v \in V_2$.		
Complete Bipartite Graph $K_{n,m}$	A bipartite graph with every possible edge		$K_{3,2}$
Star $K_{1,m}$	A Complete Bipartite start with only one vertex on one side.		$K_{1,3}$
(Hamilton) Circles C_n	Each vertex has 2 neighboors. A cycle C_n is a graph whose vertices can be arranged in a cyclic sequence, such that the edge set is $E = \{v_i v_{i+1} \| i = 1 \dots n-1\} \cup \{v_i v_n\}, n \geq 3$		$\|V\| = n$ $\|E\| = n$
Path P_n	A path P_n is a graph whose vertices can be arranged in a sequence, such that the edge set is $E = \{v_i v_{i+1} \| i = 1 \dots n-1\}$		$\|V\| = n$ $\|E\| = n-1$
Triangle	$triangle = C_3 = K_3$		$\|V\| = 3$ $\|E\| = 3$

connected graph	A graph G is connected if for every pair of distinct vertices $u, b \in V(g)$ there is a path from u to v in G.		
Tree	A tree is a connected graph without cycles. A tree is a minimal graph connecting all of its nodes.		$\lvert E\rvert = \lvert V\rvert - 1$
Disconnected graph or Subdivision	A graph with a single or multiple vertices without connection		
Regular graph	A graph G is r-regular if $\deg_g(v) = r, \forall v \in V(G)$ Paths are not regular. Complete graphs are $(n-1)$-regular		2-regular = cycles 3-regular = cubic
Plane Graph	Can be drawn without any edge crossings.		
Dual	1. rotating edges by 90° 2. replace vertices with faces and vice versa Dual of a tetrahedron is a rotated copy of itself. Dual of a Cube is a octahedron.		

Storage Formats / Data Structures

Adjacency Matrix	captions are vertices. content are edges. $A = \begin{matrix} & 1 & 2 & 3 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{matrix}, \quad A_{ij} = \begin{cases} 1, if\ ij \in E \\ 0, otherwise \end{cases}$ Note: same information as the graph Note: the sum of a row/column is the degree Undirected graphs have symetric matrices. Weighted graphs have the weights in the matrix.	place complexity: $M(n_V, n_E) \in O({n_V}^2)$ update: $O(1)$	
Adjaceny List	Each node has direct access to a list of its neighbors. Use for a sparse graph. Space-efficient.	$M(n_V, n_E) \in O(n_V + n_E)$ update: $O(\log(n))$	
Winged Edge Data Structure 1972	class Edge{Vertex X Y, Face 1 2, Edge b c d e} class Vertex{Edge[x]} class Face{Edge[x]}	simple and easily usable redundant information holes in faces not allowed	

DCEL - Doubly Connected Edge List 1978	class Vertex{Edge incident}
	class Face{Edge outer, Edge inner}
	class HalfEdge{Vertex origin,Face incident, Edge twin next prev}
	inner boundary circle is clockwise
	outer boundary circle is counter-clockwise
		very intuitive, easy to use
		restricted to surfaces
		used in CGAL
		# edges = 2 * edges
		incidence = some kind of neighboorhood
			

Quad-Edge Data Structure 1985	class Edge{Vertex origin, Edge next rot **flipped**} class Vertex{Edge **incident**} class PrimalVertex{} class DualVertex{} $Dnext = e.\underbrace{rot.rot}_{sym}.oNext.\underbrace{rot.rot}_{sym}$ $Rnext = e.rot.oNext.rot.rot.rot$ $Lnext = e.rot.rot.rot.oNext.rot$	represents map and dual can represent möbius strip simplified version in JTS # edges = 4 * edges Primal & Dual