Graph Theory: Syllabus

Week 1: Introduction [3 Lectures]

Definitions graph, vertex, vertex set, edge, edge set, adjacent vertices, edge joins vertices, multigraph, pseudograph, loop, order and size of a graph, degree of a vertex, neighbourhood of a vertex, edges incident to a vertex, isolated-vertex, end-vertex, r-regular graph, regular graph, degree sequence, minimum degree, maximum degree, isomorphic graphs, isomorphism

Theorems, Lemmas and Corollaries maximum number of edges in a graph, Handshaking Theorem, sum of vertex degrees is even, no odd number of odd degrees, two vertices of same degree, degrees of vertices of isomorphic graphs, average degree is between $\delta(G)$ and $\Delta(G)$, properties of a graph preserved by isomorphism

Week 2: Introduction (cont’d) [3 Lectures]

Definitions subgraph, induced subgraph, deletion of a vertex, spanning subgraph, complement of a graph, self-complementary graph, walk, trail, path, length of walk, cycle, closed walk, connected graphs, disconnected graphs, components, number of components, cut-vertex, bridge, complete graph, Petersen graph, hypercube, paths, cycles, bipartite graph, complete bipartite graph, union, join, Cartesian product

Theorems, Lemmas and Corollaries every walk contains a path, cycles contain two paths between pairs of vertices, bridges are not in cycles, bipartite graphs and no odd cycles

Algorithms finding a path in a walk

Week 3: Algorithms [3 Lectures]

Definitions algorithms, (worst case) complexity of an algorithm, order of a function, polynomial algorithms, searching problem, adjacency matrix, sorting problem

Theorems, Lemmas and Corollaries complexity of binary search

Algorithms sequential search algorithm, binary search algorithm, bubblesort algorithm, adjacency listing

Week 4: Trees and distance [3 Lectures]

Definitions tree, forest, binary search tree, complete m-ary tree, complete binary tree, spanning trees, fundamental cycle, directed tree, rooted tree, child, descendant, parent, ancestor, depth-first search tree, dfi($v$)

Theorems, Lemmas and Corollaries spanning trees exist if and only if graph is connected, characterizations of a tree (equivalent definitions), unique path between two vertices in a tree, each edge is a bridge in a tree, every connected graph contains a spanning tree, exact one cycle when adding an edge to a tree
**Algorithms** depth-first search algorithm

**Week 5:** Trees and distance (cont’d) [3 Lectures]

**Definitions** minimum spanning tree, breadth-first search forest, weighted graph

**Theorems, Lemmas and Corollaries** DFS characterisation of cut-vertices, DFS characterisation of bridges, Kruskal finds minimum spanning tree

**Algorithms** finding cut-vertices using DFS, finding bridges using DFS, breadth-first search algorithm Kruskal’s algorithm

**Week 6:** Trees and distance (cont’d) [2 Lectures, Good Friday]

**Definitions** distance, eccentricity of a vertex, diameter, radius, centre of a graph

**Theorems, Lemmas and Corollaries** Prim finds minimum spanning tree, distance function is a metric, \(\text{rad}(G) \leq \text{diam}(G) \leq 2 \text{rad}(G)\), centre of a tree is \(K_1\) or \(K_2\)

**Algorithms** Prim’s algorithm

**Week 7:** Trees and distance (cont’d) [1 Lecture]

**Definitions** shortest path problem

**Algorithms** Dijkstra’s algorithm

**Week 7:** Matchings and factors [2 Lectures]

**Definitions** marriage problem, assignment problem, matching, perfect matching, matched vertex, maximum matching, alternating path, augmenting path factor 1-factor 1-factorization

**Theorems, Lemmas and Corollaries** Berge’s Theorem (maximum matching if and only if no augmenting path), Tutte’s Theorem [No proof], every bridgeless cubic graph contains a perfect matching

**Algorithms** maximum matching algorithm for bipartite graphs

**Week 8:** Matchings and factors (cont’d) [3 Lectures]

**Definitions** augmentation, alternating tree, maximum alternating tree

**Theorems, Lemmas and Corollaries** Hall’s marriage theorem

**Algorithms** maximum matching algorithm for bipartite graphs

**Week 9:** Eulerian graphs [3 Lectures]

**Definitions** Eulerian trails, Eulerian circuits, Eulerian multigraphs, Chinese postman problem

**Theorems, Lemmas and Corollaries** Eulerian multigraphs and even degree vertices, Eulerian trails and at most two odd degree vertices, Chinese postman solution theorem

**Algorithms** Finding eulerian circuits

**Week 10:** Network flows [3 Lectures]
Definitions digraph (directed graph), arc, network, capacity function, capacity, flow, val $f$: value of $f$, maximum flow, cut, capacity of cut, minimum cut, $f$-saturated arc, $f$-zero arc, $f$-augmenting semipath

Theorems, Lemmas and Corollaries flow conservation theorem, comparison of val $f$ to capacity of cut, max-flow if and only if no augmenting semipath, Max-flow Min-cut Theorem

Algorithms Ford-Fulkerson algorithm

Week 11: Network flows and connectivities [3 Lectures]

Definitions $\kappa(G)$: connectivity, internal disjoint path, edge-disjoint path $\lambda(G)$: edge-connectivity

Theorems, Lemmas and Corollaries $\kappa \leq \lambda \leq \delta$, Menger’s Theorem

Algorithms labelling algorithm for network flow problem

Week 12: Colouring and planarity [3 Lectures]

Definitions vertex colouring, chromatic number, edge-colouring, edge chromatic number, planar graphs, subdivision

Theorems, Lemmas and Corollaries Brooks’ Theorem [No proof], Vizing’s Theorem [No proof], Euler’s formula, every planar graph of order $n \geq 3$ has at most $3n - 6$ edges, every planar graph contain a vertex of degree at most 5, $K_5$ and $K_{3,3}$ are nonplanar, Kuratowski’s Theorem [No proof], Four Colour Theorem [No proof]