1. Title: **Algorithmic Graph Theory**

2. **Credit Structure (L-T-P-Cr):** 3 0 2 4

3. **Course Code:** SC470

4. **Semester:** Sem V of B.Tech. (MnC), Sem VII of B.Tech. (ICT, ICT-CS), and Sem I of M.Tech. (ICT)

5. **Category:** MnC Elective/Science Elective/General Elective (Maths)

6. **Prerequisite:** Exposure to logical reasoning and proof techniques, ability to write neat computer programs, willingness to actively participate in classroom discussions.

7. **Objective/Motivation:** Graph theory finds its application in diverse fields. It has proved to be a powerful tool in proving many results in a variety of mathematical areas (also); thereby finding applications in a lot of engineering areas. Many of the proofs for graph theoretic results are constructive in nature and hence lead to finding algorithms for solving many problems and puzzles. An engineer with good knowledge of graph theory and graph algorithms is better equipped for efficiently solving a wide variety of problems. The following selection of applications of graph theory should help motivate an ICT engineer to explore more into the theory and algorithms of graphs.

   - **An application of graph theory to Computer Networks:** In ad-hoc networks cluster-based routing contribute to improved efficiency of resource use by reducing the network diameter. Consequently, many clustering algorithms have been designed for ad-hoc networks. One such design involves the modeling of the ad-hoc network as an undirected graph with the wireless nodes of the network represented by the vertices of the graph and an edge between a pair of vertices being present if the corresponding wireless nodes lie within the transmission range of one another. Under such a model, a minimal dominating set of the graph or a maximal independent set of the graph can be used as gateway set for the cluster-based routing.
An application of graph theory to Machine Intelligence: The image segmentation problem in computer vision deals with partitioning a digital image into a collection of (dis)similar pixels. The following is one of the graph theoretic approaches to solve the problem: (1) Model the image as a weighted undirected graph by associating a group of pixels with a vertex of the graph, providing an edge between vertices representing neighbouring pixels and assigning weights to the edges based on the (dis)similarity between the corresponding pixels. (2) Each tree of a minimum spanning forest of this graph will then correspond to a segment of the image.

An application of graph theory to Communication Systems: Four distinct frequency ranges are sufficient for the operation of a GSM network because of the following (a) The geographical range of the GSM network is divided into hexagonal cells. (b) The problem of range allocation in a GSM network can be modelled as a planar graph colouring problem. (c) According to the four colour theorem, no more than four colours are required to properly colour the vertices of a planar graph.

An application of graph theory to VLSI: The problem of partitioning in VLSI design is the process of decomposing a large system into independent subsystems (of manageable size) so that the subsystems can be designed concurrently. Here, the modules and the netlist are represented either as a weighted graph or as a hypergraph; the vertices represent the modules and the edges or hyperedges represent a net. The aim then is to partition the vertex set into disjoint clusters such that the cut-size and cut-cost are minimized; which is actually studied as max-flow min-cut problems in graph theory.

8. Outcomes: The student should be able to model and algorithmically solve a collection of real-life problems using graph theoretic techniques; provide proof of correctness of the (sequential, parallel, and distributed) algorithm. In the absence of efficient algorithms, (s)he should be able to provide some good working algorithms. The student should find a significant improvement in analytical and programming skill.

9. Topics (not including definitions, examples and other trivialities):

- A brief introduction to sequential, parallel, and distributed algorithms.
- A collection of results with emphasis on the following fundamental theorems and algorithms on graphs:

<table>
<thead>
<tr>
<th>Theorems</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handshaking lemmas, Havel-</td>
<td>Graph traversal algorithm(s),</td>
</tr>
<tr>
<td>Hakimi theorem, Erdos and Gallai</td>
<td>Shortest path algorithm(s),</td>
</tr>
</tbody>
</table>
theorem, The matrix-tree theorem, Cayley-Kirchoff theorem, Menger’s theorem(s), Characterization of Eulerian graphs, Dirac’s theorem, Ore’s theorem, Chvatal’s theorem, Hall’s matching condition, Konig-Egervary theorem, Tutte’s theorem, Brook’s theorem, Vizing’s theorem, Shannon’s theorem, Robertson-Seymour theorem, Kuratowski’s theorem, Five colour theorem, Max-flow min-cut theorem

Warshall’s algorithm, Prufer sequence and labeled tree construction, Minimum spanning tree algorithm(s), Fleury’s algorithm, Colouring algorithm(s), Planarity testing algorithm(s), Finding strong components, Ranking in tournaments.

- Time permitting, rudiments of extremal graph theory (Turan’s theorem, Ramsey theorem(s)) and random graph theory (Erdos-Renyi model)

10. References: Vast resources on graph theory are available today and one has an easy access to many of these either online or in our RC. The following collection is a good starting point.

- A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
- S. Even, Graph Algorithms, Computer Science Press.
- D.B. West, Introduction to Graph Theory (2nd edition), Prentice Hall of India.
- J.A. Bondy and U.S.R. Murty, Graph Theory with Applications, MacMillan Press.
- G. Agnarsson and R. Greenlaw, Graph Theory: Modelling, Applications and Algorithms, Pearson Prentice Hall.
- R. Diestel, Graph Theory (3rd edition), Graduate Texts in Mathematics Series, Springer.

11. Lab sessions: Mostly unsupervised and will involve writing computer programs for sequential/parallel/distributed graph algorithms. A short introduction will be given for each lab assignment and the students are expected to work independently on the assignment. Access to Institute’s Computing facilities for assignments requiring the same shall be provided as per the lab time-table for the course.

12. Assessment: Exam-1, (b) Exam-2, (c) Exam-3, (d) Assignments (including lab assignments), and (e) Individual project and viva-voce will be the assessment
components. Each of the four components carry 20% weightage for awarding grades. Plagiarism of any kind will lead to ‘F’ grade.

Exam-1 will be during in-semester-1 examination as announced in the Academic Calendar for Autumn 2022-23 and the syllabus shall be the topics covered until then.

Exam-2 will be during in-semester-2 examination as announced in the Academic Calendar for Autumn 2022-23 and the syllabus shall be the topics covered after Exam-1.

Exam-3 will be during end-semester examination as announced in the Academic Calendar for Autumn 2022-23 and the syllabus shall be the topics covered after Exam-2.

13. Placement, work load, illness, and other considerations: ExamTakers Category:

Registrants who do not wish to take up a course project work or who do/may not attend at least 90% of the lecture sessions or do/may not submit all the assignments on time may register into an “exam takers” list within one week of commencement of lectures for Autumn 2022-23.

Registrants who have not attended at least 90% of the lecture sessions or have not submitted some assignments on time will not be permitted to take up a course project work and will be added to the “exam takers” list.

The “exam takers” will have to participate in an additional examination as a substitute for the course project work and assignments. This exam will be conducted at the end of the course and all the topics covered shall form the syllabus for this exam.

Eventually, an “exam taker” would have taken 4 exams (Exam-1, Exam-2, Exam-3, and ExamTakers Exam) and each of these exams would carry 25% weightage for awarding grades.

14. Grading Policy: Absolute grading as per the following scheme

<table>
<thead>
<tr>
<th>Range of overall score (in %, left inclusive)</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>85 – 100</td>
<td>AA</td>
</tr>
<tr>
<td>70 – 85</td>
<td>AB</td>
</tr>
<tr>
<td>55 – 70</td>
<td>BB</td>
</tr>
<tr>
<td>40 – 55</td>
<td>BC</td>
</tr>
<tr>
<td>30 – 40</td>
<td>CC</td>
</tr>
<tr>
<td>25 – 30</td>
<td>CD</td>
</tr>
<tr>
<td>20 – 25</td>
<td>DD</td>
</tr>
<tr>
<td>15 – 20</td>
<td>DE</td>
</tr>
<tr>
<td>0 – 15</td>
<td>F</td>
</tr>
</tbody>
</table>