| | | |
|---|---|---|
| **Program** | B. Tech | **Semester-5** |
| **Type of Course** | Open Elective | |
| **Prerequisite** | Basics of Programming | |
| **Course Objective** | Students will learn to use operation research techniques to optimize resource allocation, production planning, scheduling, and logistics in real world engineering problems. They will also be able to develop a code to solve Linear Programming Problems. | |

| **Teaching Scheme** | | | | **Examination Scheme** | | | | |
|---|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Lab** | **Credit** | **Theory Marks** | | **Practical Marks** | | **Total Marks** |
| | | | | **SEE** | **CE** | **SEE** | **CE** | |
| 3 | 0 | 2 | 4 | 40 | 30 | 20 | 10 | 100 |

**List of Practical**

| | | |
|---|---|---|
| **1** | - | **List out the applications of Operation Research in IT industry and formulate the Linear Programming Problem for any one real-world problem.** |
| **2** | A | **Write a program for the given maximization Problem (Using Brut force method)**<br>Maximize:<br>$Z=3x_1+2x_2$<br>Subject to:<br>$x_1+x_2 <= 4$ and x1>=0,x2>=0<br><br>We need to find values of $x_1$ and $x_2$ that maximize Z while satisfying the constraints. |
| | B | **Implement a solution for the following problem using Brut force method.**<br>Distribute workloads across multiple servers to minimize processing time.<br><br>**Objective**: Distribute tasks across servers to minimize the **maximum processing time** on any server.<br>**Example Scenario**:<br><br>• 3 tasks with processing times: 10, 20, 30<br>• 2 servers<br><br>We want to assign tasks to servers such that the load is balanced (i.e., no server is overloaded). Hint: Approach (Simplified Brute Force) |
| **3** | | **Implement a solution for the following problem using simplex method.**<br>A factory makes **Product A** and **Product B**.<br>• Each unit of A needs:<br>    ○ 1 hours of machine time<br>    ○ 2 hours of labor<br>• Each unit of B needs:<br>    ○ 2 hours of machine time<br>    ○ 1 hour of labor<br>The factory has:<br>• **100 machine hours** available<br>• **100 labor hours** available<br>Profit:<br>• A earns ₹30 per unit<br>• B earns ₹20 per unit<br><br>**Goal:**<br>Find how many units of A and B to make to **maximize profit**, without exceeding the resource limits. |

<table>
<tr><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td>Implement a solution for the following problem using North West Corner method find a minimum cost.<br>A company has: 2 factories (suppliers) with supplies: S1 = 20, S2 = 30, 3 warehouses (consumers) with demands: D1 = 10, D2 = 25, D3 = 15<br><br>Transportation costs per unit:<br><table><tr><td></td><td>D1</td><td>D2</td><td>D3</td></tr><tr><td>S1</td><td>8</td><td>6</td><td>10</td></tr><tr><td>S2</td><td>9</td><td>7</td><td>4</td></tr></table>Note: Students are suggested to solve the above problem using Vogel’s Approximation Method.</td></tr>
<tr><td>5</td><td></td><td>Implement a solution for the following problem using Assignment Problem (Hungarian method).<br><br>Let's say there are 3 bugs and 3 developers. The following table shows the effort (cost) of each developer fixing a particular bug:<br><br><table><tr><th>Developer / Bug</th><th>Bug 1</th><th>Bug 2</th><th>Bug 3</th></tr><tr><td>Developer 1</td><td>4</td><td>2</td><td>3</td></tr><tr><td>Developer 2</td><td>2</td><td>5</td><td>6</td></tr><tr><td>Developer 3</td><td>3</td><td>7</td><td>1</td></tr></table>We want to assign the bugs to developers in such a way that the total effort (or cost) is minimized.</td></tr>
<tr><td>6</td><td></td><td>Implement a solution for the following problem using Assignment Problem.<br>The IT Project Team Formation problem involves assigning the right skills to teams to maximize the project's overall performance. The challenge is to allocate team members to various tasks based on their skills in such a way that the project’s performance is maximized.<br><br>This problem can be formulated as an assignment problem where:<br>The tasks are the "jobs" that need to be done.<br>The team members are the "agents" who can perform these tasks.<br>The performance (effort or cost) of each team member performing a task is given by a matrix.<br><br>The goal is to assign each team member to the right task in such a way that the overall project performance is maximized.<br><br>Input:<br><br>The input matrix represents the performance of each team member on each task. The rows represent team members, and the columns represent tasks. Each element in the matrix shows the performance score of a team member on a specific task.</td></tr>
<tr><td>7</td><td></td><td>Implement a solution for CPU Scheduling Problem: Minimizing Waiting and Turnaround Time<br>You are given multiple processes with their burst times (execution times). Your task is to assign CPU time to each process so that the average waiting time and average turnaround time are minimized.<br>Hint:<br>Implement a SJF algorithm<br>Burst Time: Time required by a process for execution.<br>Waiting Time: Time a process waits in the ready queue.<br>Waiting Time=Turnaround Time−Burst Time<br>Turnaround Time: Total time taken from arrival to completion.<br>Turnaround Time=Completion Time−Arrival Time</td></tr>
<tr><td>8</td><td></td><td>Implement Two-Person Zero-Sum Game using a Saddle Point (Pure Strategy)<br>In Game Theory, a Two-Person Zero-Sum Game is a situation where:<br>Two players (Player A and Player B) play a game.</td></tr>
</table>

<table>
<tr><td></td><td></td><td>The gain of one player is the loss of the other.<br>The sum of gains and losses is always zero.<br><br>The goal is to find an optimal strategy for each player such that the expected payoff for both players is maximized or minimized depending on their role (row or column player).<br><br>A simple way to solve such games is by checking if a saddle point exists:<br>A saddle point is the element that is minimum in its row and maximum in its column.<br>If a saddle point exists, the game has a pure strategy solution and find it's solution<br>Otherwise, find a solution using arithmetic/algebraic method<br><br>Input: 2 X 2 matrix<br>Output: Saddle point value / find the value of game (maximum winning)</td></tr>
<tr><td>9</td><td></td><td>Implement a PERT-CPM based solution for following problem.<br>PERT (Program Evaluation Review Technique) and CPM (Critical Path Method) are used in project scheduling to: Estimate the minimum time required to complete a project, Identify the critical path, i.e., the longest path through the network which determines the project duration.<br><br>A project has 6 activities labeled A to F:<br>Activity Duration Predecessor<br>A 3 -<br>B 2 A<br>C 4 A<br>D 2 B, C<br>E 3 C<br>F 1 D, E<br><br>Write a program that calculates Earliest Start (ES), Earliest Finish (EF), Latest Start (LS), Latest Finish (LF), Slack Time, Critical Path</td></tr>
</table>