UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
TIPO DE EXAMEN Y/O EVALUACIÓN:
MEDIO CURSO (*Midterm Exam*)

MATERIA/UNIDAD DE APRENDIZAJE: Temas Selectos de Optimización
LEARNING UNIT: Selected Topics on Optimización (in English)

SEMESTER: January – June 2026
ACADEMY: Statistics and Operations Research (*Estadística e Investigación de Operaciones*).
INSTRUCTOR: Dr. Roger Z. Ríos Mercado

DIRECTIONS.- Answer the following questions and/or exercises in the answer sheet. Do not write in this sheet

SECTION 1: QUESTIONS (50 POINTS)

Answer and justify your answer.

1. [*UT1: Combinatorial optimization*; 6 pts] Define a combinatorial optimization problem.
2. [*UT1: Combinatorial optimization*; 6 pts] Define and explain what a brute-force enumeration method is for solving a combinatorial optimization problem.
3. [*UT1: Combinatorial optimization*; 6 pts] When do we say that an optimization problem is easy to solve.
4. [*UT2: Heuristics*; 6 pts] Define and explain what a heuristic method is for solving combinatorial optimization problems.
5. [*UT2: Heuristics*; 6 pts] Under what circumstances or when it is preferable to use a heuristic method instead of an exact optimization method for solving a combinatorial optimization problem?
6. [*UT2: Constructive heuristics*; 6 pts] What is a constructive heuristic?
7. [*UT2: Constructive heuristics for the TSP*; 7 pts] Describe in detail the nearest neighbor heuristic for solving the Traveling Salesman Problem. You may illustrate your idea with an example or drawing.
8. [*UT2: Constructive heuristics for the TSP*; 7 pts] Describe in detail the nearest insertion heuristic for solving the Traveling Salesman Problem. You may illustrate your idea with an example or drawing.

SECTION 2: PROBLEMS (50 POINTS)

9. We have a matching problem (MP) defined as follows. Given a graph G = (V, E), where V is the set of nodes or vertices and E is the set of edges, and given a weight w_{ij} for each edge (i,j) in E, we must find a perfect matching M whose weight sum $w(M) = \sum_{(i,j)\in M} w_{ij}$ is as large as possible.

   A perfect matching M is defined as s subset of edges from G with the property that each node i in V is incident to an edge in M, that is, M = {(i,j) $\in$ E : there are no two or more edges sharing the same node}. Figure 1 shows a graph G = (V, E), with V = {1, 2, …, 12} and edge set E indicated in the picture. Figure 2 illustrates a perfect matching M given by M = {(1,5), (2,3), (6,7), (4,8), (9,10), (11,12)}. Figure 3 shows an edge subset M_2, which is not a matching because edges (1,5) and (5,6) are incident to a same vertex (node 5).

   (a) [*UT1: Combinatorial optimization*; 6 pts] Is M^1 = {(1,2), (3,8), (7,12), (5,9), (10,11)} a feasible solution? Justify your answer.

   (b) [*UT1: Combinatorial optimization*; 6 pts] Is M^2 = {(7,10), (11,12), (4,8), (2,3), (1,5), (6,9)} a feasible solution? Justify your answer.

   (c) [*UT1: Combinatorial optimization*; 8 pts] Sort the following three solutions from best to worst. Justify your answer.
   M^3 = {(1,5), (2,6), (3,7), (4,8), (7,11), (9,10)},
   M^4 = {(1,2), (5,9), (6,10), (3,7), (4,8), (11,12)},
   M^5 = {(1,5), (2,6), (9,10), (3,4), (7,11), (8,12)}.

   (d) [*UT2: Constructive heuristics*; 20 pts] Design a constructive heuristic for this MP. Show clearly each step either in pseudocode or flow chart.

   (e) [*UT2: Constructive heuristics*;10 pts] Illustrate how your heuristic works, step by step, by applying it to the example below.

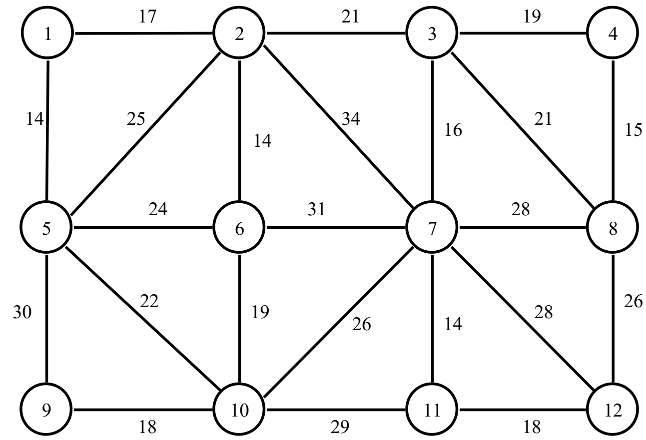

Figure 1: Original graph. Each node (circle) is identified by a label shown inside of it. In each edge (*i,j*), weight w_{ij} is indicated.

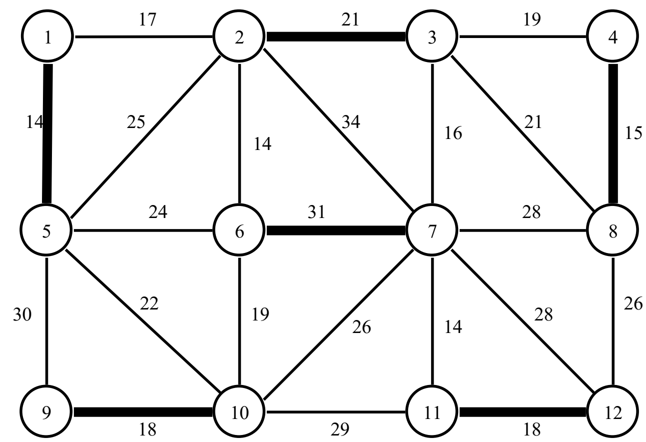

Figure 2: Example of a perfect matching M. in this case, M (denoted by bold edges) is given by M = {(1,5), (2,3), (6,7), (4,8), (9,10), (11,12)}. Its total weight is given by $w(M) = w_{1,5} + w_{2,3} + w_{6,7} + w_{4,8} + w_{9,10} + w_{11,12} = 117$.

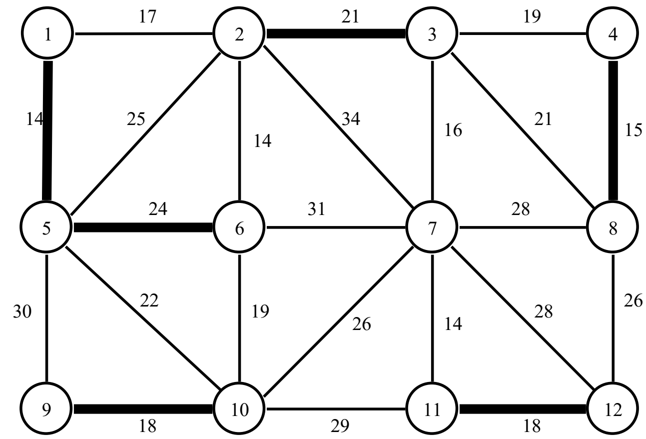

Figure 3: In this case, M_2 (denoted by bold edges) is not a matching because edges (1,5) and (5,6) share a common node (node 5).