//amsmath, amssymb

Operations Research: Models and Algorithms
Homework 1B

B12508023

December 4, 2025

Problem 1: Distribution Center Location Problem

(a) Integer Programming Formulation (Unlimited Capacity)

Decision Variables:

- $x_{ij} \in \mathbb{R}_{\geq 0}$: Number of units shipped from location j to city i
- $y_j \in \{0, 1\}$: 1 if a distribution center is opened at location j, 0 otherwise

Parameters:

- H_i: Demand of city i
- C_j: Weekly operating cost of location j (in \$)
- S_{ij}: Unit shipping cost from location j to city i

Objective:

$$\min \sum_j C_j y_j + \sum_i \sum_j S_{ij} x_{ij}$$

Subject to:

$$\begin{aligned}
&\text{(1) Demand satisfaction:} && \sum_j x_{ij} = H_i \quad \forall i \\
&\text{(2) Ship only from opened centers:} && x_{ij} \leq H_i \cdot y_j \quad \forall i, j \\
&\text{(3) Variable domains:} && x_{ij} \geq 0 \quad \forall i, j; \quad y_j \in \{0, 1\} \quad \forall j
\end{aligned}$$

(b) Integer Programming Formulation (With Capacity Constraints)

Additional Parameter:

- K_j: Weekly capacity of the distribution center at location j

Objective: (same as part (a))

$$\min \sum_j C_j y_j + \sum_i \sum_j S_{ij} x_{ij}$$

Subject to:

(1) Demand satisfaction: $$\sum_j x_{ij} = H_i \quad \forall i$$

(2) Ship only from opened centers: $$x_{ij} \leq H_i \cdot y_j \quad \forall i, j$$

(3) Capacity constraint: $$\sum_i x_{ij} \leq K_j \cdot y_j \quad \forall j$$

(4) Variable domains: $$x_{ij} \geq 0 \quad \forall i, j; \quad y_j \in \{0, 1\} \quad \forall j$$

(c) Optimal Solution for the Instance (Unlimited Capacity)

Table 1: City Data: Weekly Demand and Coordinates

City	Demand	X	Y
1	50000	52	68
2	16000	12	70
3	33000	98	1
4	43000	86	40
5	48000	46	27
6	29000	16	96
7	45000	47	97
8	19000	68	58
9	40000	73	49
10	23000	46	79
11	25000	14	44
12	26000	83	70
13	38000	65	65
14	26000	55	95
15	28000	18	91
16	20000	92	21
17	45000	65	68
18	35000	4	56
19	14000	52	50
20	28000	40	81

optimal_distribution_network.png

Figure 1: Optimal distribution network based on Euclidean distance. Shipping cost is computed as $S_{ij} = S \cdot \sqrt{(X_i^C - X_j^L)^2 + (Y_i^C - Y_j^L)^2}$. The country is 100 km $\times$ 100 km in size.

Problem 2

(a) Pseudocode for Primality Test

We define the following pseudocode to determine whether a given positive integer a is a prime number:

```
Function is_prime(a):
    If a <= 1 then
        Return False
    For i from 2 to floor sqrt(a) do
        If a mod i == 0 then
            Return False
    Return True

Input a
Output is_prime(a)
```

(b) Time Complexity Analysis

The algorithm checks divisibility of the input a by all integers from 2 to $\lfloor \sqrt{a} \rfloor$.

Each iteration performs a single modulus operation, and there are at most $\sqrt{a}$ iterations.

Therefore, the worst-case time complexity is:

$$\boxed{O(\sqrt{a})}$$

(c) Gaussian Elimination for a Linear System

Given the system:

$$\begin{cases} 2x_2 + 2x_3 = 4 \\ x_1 + 2x_2 + x_3 = 4 \\ x_2 - x_3 = 0 \end{cases}$$

We rewrite this in augmented matrix form:

$$\left[\begin{array}{ccc|c} 0 & 2 & 2 & 4 \\ 1 & 2 & 1 & 4 \\ 0 & 1 & -1 & 0 \end{array}\right]$$

Swap row 1 and row 2:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 2 & 2 & 4 \\ 0 & 1 & -1 & 0 \end{array}\right]$$

Eliminate below pivot:

$$R_2 \leftarrow R_2 - 2 \cdot R_3 \Rightarrow \left[\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 0 & 4 & 4 \\ 0 & 1 & -1 & 0 \end{array}\right]$$

Back-substitution gives:

$$x_3 = 1, \quad x_2 = 1, \quad x_1 = 1$$

Conclusion: The system has a **unique solution:**

$$\boxed{x_1 = 1, \quad x_2 = 1, \quad x_3 = 1}$$

(d) Finding the Inverse of a Matrix via Gaussian Elimination

Given:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 2 \\ 3 & 1 & 1 \end{bmatrix}$$

Form the augmented matrix $[A \mid I]$:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & 1 & 2 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Row operations:

$$\begin{aligned} & R_2 \leftarrow R_2 - 4R_1 \\ & R_3 \leftarrow R_3 - 3R_1 \\ \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & -4 & 1 & 0 \\ 0 & 1 & 1 & -3 & 0 & 1 \end{array}\right] & \\ & R_3 \leftarrow R_3 - R_2 \\ \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & -4 & 1 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 \end{array}\right] & \\ & R_3 \leftarrow -1 \cdot R_3 \\ & R_2 \leftarrow R_2 - 2 \cdot R_3 \end{aligned}$$

Final result:

$$A^{-1} = \boxed{\begin{bmatrix} 1 & 0 & 0 \\ -2 & -1 & 2 \\ -1 & 1 & -1 \end{bmatrix}}$$