Ankit Jainar
EE24BTECH11004
IIT Hyderabad.

November 6, 2024

1. Problem

2. Solution
   - Section Formula
   - Ratio's
   - Point's
   - Plot

# Problem Statement

Find the coordinates of the points which divide the line segment joining A(-2, 2) and B(2, 8) into four equal parts.

# Section Formula

The section formula for internal division, the coordinates of the point dividing the line in the ratio $k : 1$ are given by:

$$R_k = \left( \frac{k \cdot x_2 + x_1}{k+1}, \frac{k \cdot y_2 + y_1}{k+1} \right) \tag{3.1}$$

# Ratio's

$k = \frac{i}{n-i}\ n$, $0 < i < n$ is number of equal parts
For $n = 4$
now for

$$R_1, k = \frac{1}{3} \tag{3.2}$$

$$R_2, k = 1 \tag{3.3}$$

$$R_3, k = 3 \tag{3.4}$$

# Point's

substituting A=$(-2, 2)$ and B=$(2, 8)$ in $R_k$ we get

$$R_1 = (-1.0, 3.5) \tag{3.5}$$
$$R_2 = (0.0, 5.0) \tag{3.6}$$
$$R_3 = (1.0, 6.5) \tag{3.7}$$

# Plot

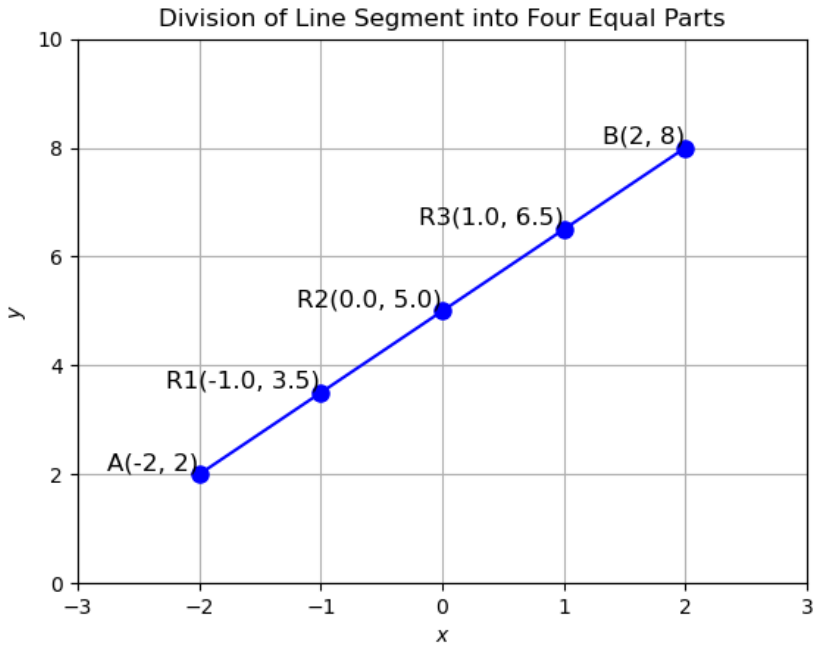


Figure: Stem Plot of y($n$)

# C code

```c
#include <stdio.h>

// Define a structure for a point
typedef struct {
    float x;
    float y;
} Point;

// Function to divide the line segment into n equal parts
void divide_line(Point A, Point B, int n, FILE *file) {
    fprintf(file, "Dividing into %d parts:\n", n);
    for (int i = 1; i < n; i++) {
        float x = (i * B.x + (n - i) * A.x) / n;
        float y = (i * B.y + (n - i) * A.y) / n;
        fprintf(file, "%.2f %.2f\n", x, y);  // Write the point to the file
    }
    fprintf(file, "\n");
}

int main() {
    // Define points A and B
    Point A = {-2, 2};
    Point B = {2, 8};

    // Open a file to store the output points
    FILE *file = fopen("points.txt", "w");

    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
    }

    // Divide the line for different values of k and write the results to the file
    divide_line(A, B, 3, file);  // Divide into 3 parts
    divide_line(A, B, 4, file);  // Divide into 4 parts
    divide_line(A, B, 5, file);  // Divide into 5 parts

    // Close the file
    fclose(file);

    printf("Points computed and written to 'points.txt'.\n");
    return 0;
}

```

# Python code

```python
import matplotlib.pyplot as plt
# Function to find points dividing a line segment into n equal parts
def divide_line(A, B, n):
    points = []
    for i in range(1, n):
        x = (i * B[0] + (n - i) * A[0]) / n
        y = (i * B[1] + (n - i) * A[1]) / n
        points.append((x, y))
    return points

# Points A and B
A = (-2, 2)
B = (2, 8)

# Divide the line into 4 equal parts
dividing_points = divide_line(A, B, 4)

# Print the dividing points
print("Dividing points:")
for i, point in enumerate(dividing_points, start=1):
    print(f"R{i}: {point}")

# Add points A and B to the list for plotting
points = [A] + dividing_points + [B]

# Create a figure and axis
fig, ax = plt.subplots()

# Extract x and y values
x_values = [point[0] for point in points]
y_values = [point[1] for point in points]

# Plot the line segment and points
ax.plot(x_values, y_values, marker='o', color='blue', linestyle='-', markersize=8)

# Add labels to the points
ax.text(A[0], A[1], 'A(-2, 2)', fontsize=12, ha='right', va='bottom')
for i, point in enumerate(dividing_points, start=1):
    ax.text(point[0], point[1], f'R{i}{point}', fontsize=12, ha='right', va='bottom')
ax.text(B[0], B[1], 'B(2, 8)', fontsize=12, ha='right', va='bottom')
# Set limits and grid
ax.set_xlim(-3, 3)
ax.set_ylim(0, 10)
ax.grid(True)
# Set labels for axes
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
# Set title
plt.title('Division of Line Segment into Four Equal Parts')
# Show the plot
plt.show()

```