2.4.33

M Chanakya Srinivas- EE25BTECH11036

Problem

Name the type of triangle formed by the points

$$A(-5,6), \quad B(-4,-2), \quad C(7,5).$$

Vertices

$$\mathbf{A} = \begin{pmatrix} -5 \\ 6 \end{pmatrix},$$
$$\mathbf{B} = \begin{pmatrix} -4 \\ -2 \end{pmatrix},$$
$$\mathbf{C} = \begin{pmatrix} 7 \\ 5 \end{pmatrix} \tag{1}$$

Difference Vectors

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} 1 \\ -8 \end{pmatrix}, \tag{2}$$

$$\mathbf{C} - \mathbf{A} = \begin{pmatrix} 12 \\ -1 \end{pmatrix}, \tag{3}$$

$$\mathbf{C} - \mathbf{B} = \begin{pmatrix} 11 \\ 7 \end{pmatrix}. \tag{4}$$

Angle at A

$$(\mathbf{B}-\mathbf{A})^{\top}(\mathbf{C}-\mathbf{A}) = \begin{pmatrix} 1 & -8 \end{pmatrix} \begin{pmatrix} 12 \\ -1 \end{pmatrix}$$
$$= 12 + 8 = 20 > 0 \tag{5}$$

So, $\angle A$ is acute.

Angle at B

$$(\mathbf{A}-\mathbf{B})^{\top}(\mathbf{C}-\mathbf{B}) = \begin{pmatrix} -1 & 8 \end{pmatrix} \begin{pmatrix} 11 \\ 7 \end{pmatrix}$$
$$= -11 + 56 = 45 > 0 \tag{6}$$

So, $\angle B$ is acute.

Angle at C

$$(\mathbf{A}-\mathbf{C})^{\top}(\mathbf{B}-\mathbf{C}) = \begin{pmatrix} -12 & 1 \end{pmatrix} \begin{pmatrix} -11 \\ -7 \end{pmatrix}$$
$$= 132 - 7 = 125 > 0 \tag{7}$$

So, $\angle C$ is acute.

Conclusion

All dot products are positive $\Rightarrow$ all angles are acute. Also, side lengths are unequal $\Rightarrow$ scalene.

$$\boxed{\text{The triangle is an acute scalene triangle.}}$$

C Code

```c
#include <stdio.h>
#include <math.h>

// Function to compute squared distance using matrices
double dist2(double P[2], double Q[2]) {
    double diff[2];
    diff[0] = P[0] - Q[0];
    diff[1] = P[1] - Q[1];
    return diff[0]*diff[0] + diff[1]*diff[1];
}

int main() {
    double A[2] = {-5, 6};
    double B[2] = {-4, -2};
    double C[2] = {7, 5};

    // Distances squared
    double AB2 = dist2(A, B);
    double BC2 = dist2(B, C);
```

C Code

```
    double CA2 = dist2(C, A);

    printf(AB^2 = %.2lf, BC^2 = %.2lf, CA^2 = %.2lf\n, AB2, BC2,
        CA2);

    // Check for type
    if (fabs(AB2 - BC2) < 1e-6 || fabs(BC2 - CA2) < 1e-6 || fabs(
        CA2 - AB2) < 1e-6)
        printf(The triangle is Isosceles.\n);
    else
        printf(The triangle is Scalene.\n);

    if (fabs(AB2 + BC2 - CA2) < 1e-6 || fabs(BC2 + CA2 - AB2) < 1
        e-6 || fabs(CA2 + AB2 - BC2) < 1e-6)
        printf(It is also a Right-angled triangle.\n);

    return 0;
}
```

Only Python Code

```
import sys
sys.path.insert(0, '/home/chanakya/MATGEO/2.4.33/codes') # path
    to local scripts
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

# local imports
from libs.line.funcs import *

# Function to compute squared distance
def dist2(P, Q):
    return LA.norm(P-Q)**2

# Given points
A = np.array([-5, 6]).reshape(-1,1)
B = np.array([-4, -2]).reshape(-1,1)
C = np.array([7, 5]).reshape(-1,1)
```

Only Python Code

```
# Distances squared
AB2 = dist2(A,B)
BC2 = dist2(B,C)
CA2 = dist2(C,A)

# Determine type
triangle_type = []
if np.isclose(AB2, BC2) or np.isclose(BC2, CA2) or np.isclose(CA2
    , AB2):
    triangle_type.append(Isosceles)
else:
    triangle_type.append(Scalene)

if (np.isclose(AB2+BC2, CA2) or
    np.isclose(BC2+CA2, AB2) or
    np.isclose(CA2+AB2, BC2)):
    triangle_type.append(Right-angled)

triangle_type =  .join(triangle_type)
```

Only Python Code

```
# Generate triangle sides
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)

# Plotting
plt.plot(x_AB[0,:], x_AB[1,:], label='$AB$')
plt.plot(x_BC[0,:], x_BC[1,:], label='$BC$')
plt.plot(x_CA[0,:], x_CA[1,:], label='$CA$')

# Labeling the coordinates
tri_coords = np.block([[A,B,C]])
vert_labels = ['A','B','C']
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}({int(tri_coords[0,i])},{int(tri_coords
        [1,i])})',
                 (tri_coords[0,i], tri_coords[1,i]),
                 textcoords=offset points,
                 xytext=(20,5), ha='center')
```

Only Python Code

```
# Show triangle type at centroid
centroid = (A+B+C)/3
plt.text(centroid[0,0], centroid[1,0],
        f{triangle_type} Triangle,
        fontsize=12, color='green',
        bbox=dict(facecolor='yellow', alpha=0.3, edgecolor='black
            '))

# Axes formatting
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')

plt.grid()
```

Only Python Code

```
plt.axis('equal')
plt.legend(loc='best')
plt.title(Triangle ABC with Classification)
plt.show()
```

Python Code with Shared Output

```
import numpy as np
import matplotlib.pyplot as plt

# Points (convert to numpy int to ensure matrix ops but cast for
    display)
A = np.array([-5, 6])
B = np.array([-4, -2])
C = np.array([7, 5])

# Matrix difference and squared distance
def dist2(P, Q):
    diff = P - Q
    return diff @ diff # dot product

# Distances squared
AB2 = dist2(A, B)
BC2 = dist2(B, C)
CA2 = dist2(C, A)
```

Python Code with Shared Output

```
# Determine type
triangle_type = []
if np.isclose(AB2, BC2) or np.isclose(BC2, CA2) or np.isclose(CA2
    , AB2):
    triangle_type.append(Isosceles)
else:
    triangle_type.append(Scalene)

if (np.isclose(AB2 + BC2, CA2) or
    np.isclose(BC2 + CA2, AB2) or
    np.isclose(CA2 + AB2, BC2)):
    triangle_type.append(Right-angled)

triangle_type =  .join(triangle_type)

print(fAB^2 = {AB2}, BC^2 = {BC2}, CA^2 = {CA2})
print(fThe triangle is {triangle_type}.)

# ---- Plotting ----
```

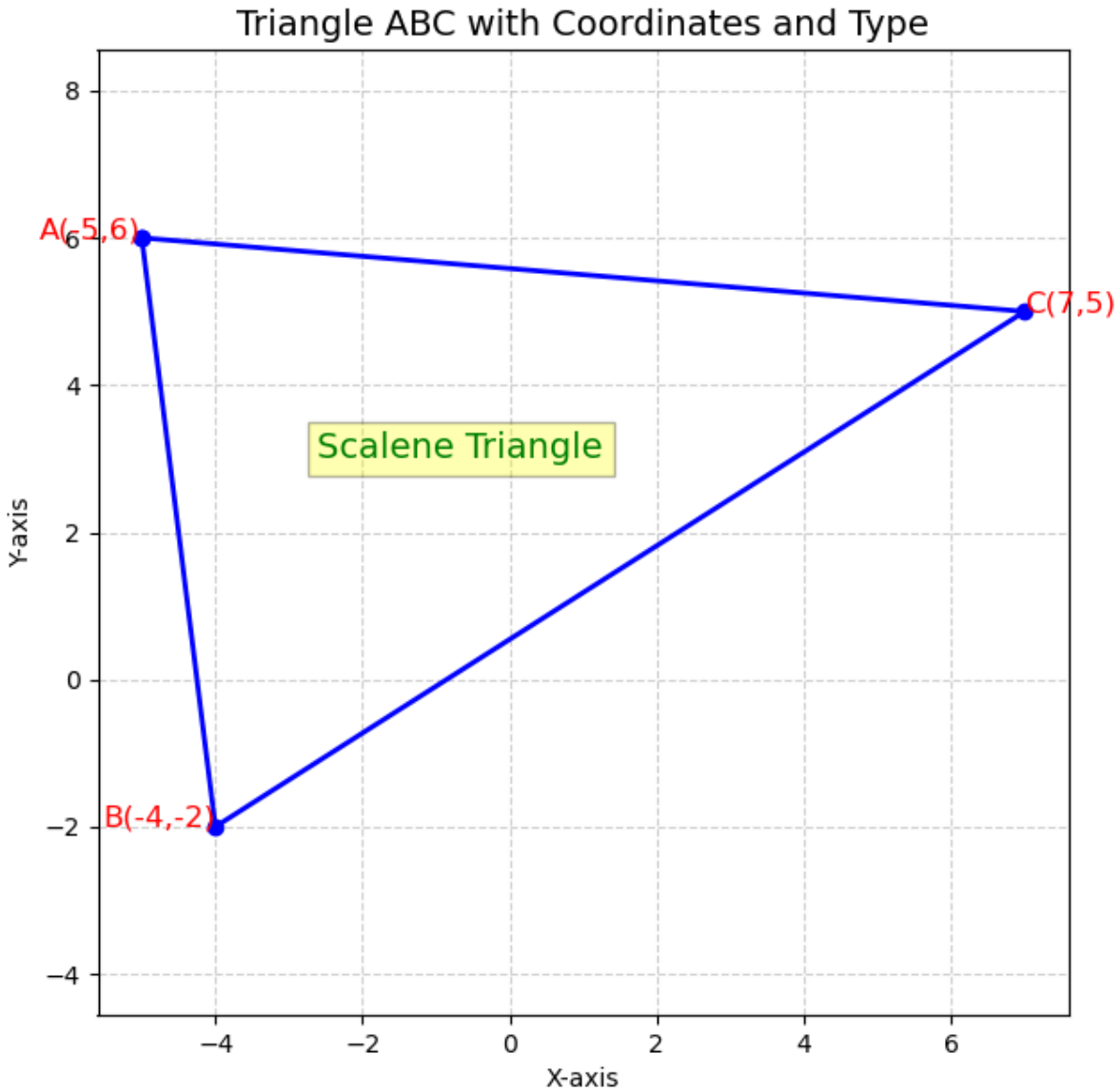
Python Code with Shared Output

```
plt.figure(figsize=(7,7))
plt.plot(X, Y, 'b-o', linewidth=2)

# Annotate points with pure int coordinates
plt.text(A[0], A[1], fA({int(A[0])},{int(A[1])}), fontsize=12,
    color='red', ha='right')
plt.text(B[0], B[1], fB({int(B[0])},{int(B[1])}), fontsize=12,
    color='red', ha='right')
plt.text(C[0], C[1], fC({int(C[0])},{int(C[1])}), fontsize=12,
    color='red', ha='left')

# Show triangle type inside the plot
plt.text((A[0]+B[0]+C[0])/3, (A[1]+B[1]+C[1])/3,
        f{triangle_type} Triangle,
        fontsize=14, color='green', ha='center',
        bbox=dict(facecolor='yellow', alpha=0.3, edgecolor='black
            '))

plt.grid(True, linestyle=--, alpha=0.6)
```

Python Code with Shared Output

```
plt.axis(equal)
plt.title(Triangle ABC with Coordinates and Type, fontsize=14)
plt.xlabel(X-axis)
plt.ylabel(Y-axis)
plt.show()
```

Python plot using shared output



only Python plot