1.5.35

Revanth Siva Kumar D- EE25BTECH11048

Question

Question:The mid-point of segment AB is the point $P(0, 4)$. If the coordinates of B are $(-2, 3)$ then the coordinates of A are _____.

Theoretical Solution

Given Information

The midpoint of segment AB is $P(0, 4)$.

The coordinates of point B are $(-2, 3)$.

We need to find the coordinates of point A using a specific matrix method based on the section formula.

Matrix Setup

First, write the coordinates of the points as column matrices:

$$P = \begin{pmatrix} 0 \\ 4 \end{pmatrix},$$

$$B = \begin{pmatrix} -2 \\ 3 \end{pmatrix},$$

$$A = \begin{pmatrix} x \\ y \end{pmatrix}$$

Theoretical Solution

The Formula

Since P is the midpoint, it is known that A divides BP in the ratio -2:1 internally or in other words 2:1 externally. Here k = -2, Thus by section formula:

$$A = \frac{kP + B}{1 + k}$$

Substituting k = -2 we get

$$A = 2P - B$$

Calculation

Substitute the matrices:

$$A = 2\begin{pmatrix} 0 \\ 4 \end{pmatrix} - \begin{pmatrix} -2 \\ 3 \end{pmatrix}$$

Scalar multiplication:

$$A = \begin{pmatrix} 0 \\ 8 \end{pmatrix} - \begin{pmatrix} -2 \\ 3 \end{pmatrix}$$

Theoretical Solution

Matrix subtraction:

$$A = \begin{pmatrix} 0-(-2) \\ 8-3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

Conclusion

The coordinates of point A are $(2, 5)$.

Quick check: midpoint of $A(2, 5)$ and $B(-2, 3)$ is

$$\left(\frac{2+(-2)}{2}, \frac{5+3}{2} \right) = (0, 4) = P$$

C Code - Section formula function

```c
#include <stdio.h>

void findA(int xp, int yp, int xb, int yb, int *xa, int *ya) {
    *xa = 2*xp - xb;
    *ya = 2*yp - yb;
}

int main() {
    int xp=0, yp=4;
    int xb=-2, yb=3;
    int xa, ya;

    findA(xp, yp, xb, yb, &xa, &ya);

    printf(Coordinates of A: (%d, %d)\n, xa, ya);
    return 0;
}
```

Python Code through shared output

```
import matplotlib.pyplot as plt

def findA(xp, yp, xb, yb):
    xa = 2*xp - xb
    ya = 2*yp - yb
    return xa, ya


xp, yp = 0, 4
xb, yb = -2, 3

xa, ya = findA(xp, yp, xb, yb)
print(fCoordinates of A: ({xa}, {ya}))


plt.figure(figsize=(6,6))
plt.scatter([xa, xb, xp], [ya, yb, yp], color=['red','blue','
    green'], s=100)
```

Python Code through shared output

```
plt.text(xa+0.1, ya, A(2,5), fontsize=12)
plt.text(xb+0.1, yb, B(-2,3), fontsize=12)
plt.text(xp+0.1, yp, P(0,4), fontsize=12)


plt.plot([xa, xb], [ya, yb], 'k--', label=AB)
plt.scatter(xp, yp, color='green', s=120, marker='x', label=
    Midpoint P)

plt.axhline(0, color='gray', linewidth=0.5)
plt.axvline(0, color='gray', linewidth=0.5)
plt.legend()
plt.grid(True)
plt.show()
```

Python code : Direct

```
import sys
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

#local imports
from libs.line.funcs import *
from libs.triangle.funcs import *
from libs.conics.funcs import circ_gen


#if using termux
import subprocess
import shlex
#end if
```

Python code : Direct

```
#Given points
P = np.array(([0,4])).reshape(-1,1)
B = np.array(([-2,3])).reshape(-1,1)

#Ratio
n=-2/1

#Point
A= (B+n*P)/(1+n) # calculating the coordinate points of R which
    divides the join between the two points
#print(R)

#Generating all lines
x_PB = line_gen(A,B)

#Plotting all lines
plt.plot(x_PB[0,:],x_PB[1,:],label='$PB$')
```

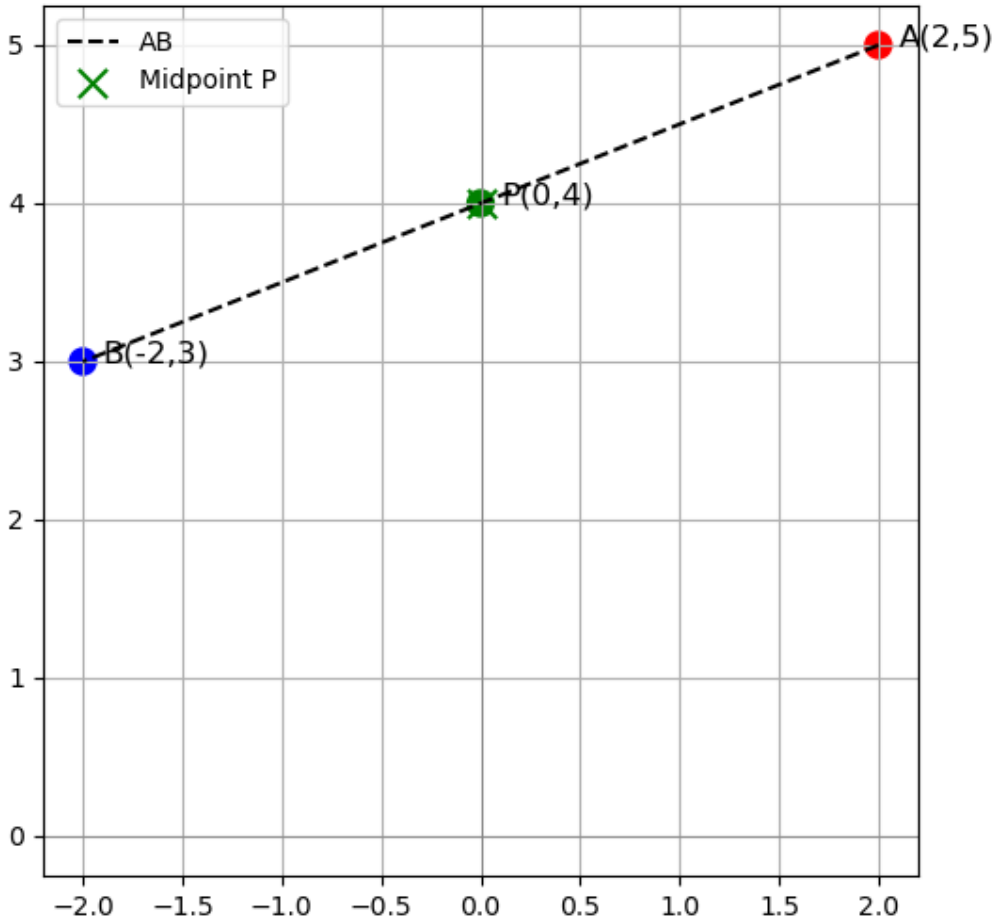
Python code : Direct

```
#Labeling the coordinates
tri_coords = np.block([[P,B,A]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['P','B','A']
for i, txt in enumerate(vert_labels):
    #plt.annotate(txt, # this is the text
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i
        ]:.0f})',
                 (tri_coords[0,i], tri_coords[1,i]), # this is the
                      point to label
                 textcoords=offset points, # how to position the
                     text
                 xytext=(20,-10), # distance from text to points (
                     x,y)
                 ha='center') # horizontal alignment can be left,
                     right or center
# use set_position
```

Python code : Direct

```
ax = plt.gca()
#ax.spines['top'].set_color('none')
#ax.spines['left'].set_position('zero')
#ax.spines['right'].set_color('none')
#ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
#plt.xlabel('$x$')
#plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid() # minor
plt.axis('equal')
plt.show()
```

Plot by python using shared output from c



Plot by python using shared output from c