

1.5.35

Revanth Siva Kumar D- EE25BTECH11048

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

```
#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;
}
```

Python Code through shared output

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

# Load the compiled C library
lib = ctypes.CDLL('./findA.so')

# Define the function signature: void findA(int, int, int, int,
    int*, int*)
lib.findA.argtypes = [
    ctypes.c_int, ctypes.c_int,
    ctypes.c_int, ctypes.c_int,
    ctypes.POINTER(ctypes.c_int),
    ctypes.POINTER(ctypes.c_int)
]
lib.findA.restype = None # void
```

Python Code through shared output

```
def findA(xp, yp, xb, yb):  
    Call the C function via ctypes.  
    xa = ctypes.c_int()  
    ya = ctypes.c_int()  
    lib.findA(xp, yp, xb, yb, ctypes.byref(xa), ctypes.byref(ya))  
    return xa.value, ya.value  
  
if __name__ == __main__:  
    xp, yp = 3, 4  
    xb, yb = 1, 2  
  
    xa, ya = findA(xp, yp, xb, yb)  
    print(fCoordinates of A (from C): ({xa}, {ya}))
```


Python Code through shared output

```
# Plotting
plt.figure(figsize=(6, 6))
plt.plot(xp, yp, 'ro', label='P')
plt.plot(xb, yb, 'bo', label='B')
plt.plot(xa, ya, 'go', label='A')
plt.plot([xb, xa], [yb, ya], 'k--', label='Line BA')

plt.text(xp+0.1, yp+0.1, P)
plt.text(xb+0.1, yb+0.1, B)
plt.text(xa+0.1, ya+0.1, A)

plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(True)
plt.legend()
plt.title(Reflection of B across P (computed by C, plotted by
          Python))
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

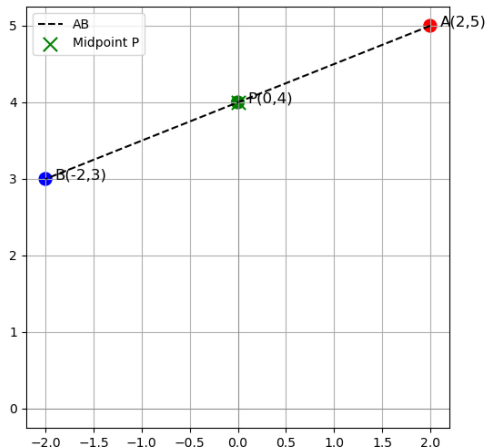
```
#Labeling the coordinates
tri_coords = np.block([[P,B,A]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['P','B','A']
for i, txt in enumerate(vert_labels):
    #plt.annotate(txt, # this is the text
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]
        ]:.0f})',
        (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

