

Ejercicios de Secuencia Lógica y Funciones

13 de julio de 2023

Emplear los casos de prueba para identificar con precisión el requerimiento, así como utilizar las salidas esperadas para revisar el correcto funcionamiento de la solución propuesta. Se sugiere a manera de ejercicio no utilizar condicionales en las soluciones en caso de tener conocimientos más avanzados de programación, resolver únicamente mediante secuencia lógica utilizando operadores básicos con su jerarquía de signos, y los operadores: `//`, `%`, `max`, `min`, `len` y `sum`.

1. Escribir una función que reciba las componentes de tres puntos cartesianos en el siguiente orden: $x_A, y_A, x_B, y_B, x_C, y_C$, y retorne la menor distancia que hay al comparar la distancia desde P_A hasta P_B y desde P_A hasta P_C , recordando que el cálculo de la distancia entre dos puntos $P_A(x_A, y_A)$ y $P_B(x_B, y_B)$ está dada por la Ecuación 1 presentada a continuación:

$$d_{P_A, P_B} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (1)$$

No utilizar librerías adicionales en la solución, únicamente con las herramientas de **Python** utilizadas en clase, con el fin de estandarizar las pruebas que se realizarán, y recordar que por propiedades de potencias, una raíz cuadrada equivale a elevar a potencia racional o fraccionaria de $1/2$, como se muestra en la Ecuación 2.

$$\sqrt{x} = x^{\frac{1}{2}} \quad (2)$$

Es importante anotar que se espera que la implementación retorne solamente un valor, es decir, la distancia menor encontrada, y que las distancias no deben tener signo negativo.

A continuación los casos de prueba que describen el comportamiento esperado de la función:

Detalle de Casos:

```
[{'Entrada': (0, 0, 3, 4, -7, 1), 'Resultado Esperado': 5.0},
 {'Entrada': (1, 1, 2, 2, 1, 0), 'Resultado Esperado': 1.0},
 {'Entrada': (0, 0, 6, 3, -3, 2), 'Resultado Esperado': 3.605551275463989},
 {'Entrada': (1, 1, 1, 1, 1, 1), 'Resultado Esperado': 0.0},
 {'Entrada': (0, 0, 0, 0, 1, 1), 'Resultado Esperado': 0.0},
 {'Entrada': (0, 0, 6, 6, -6, -6), 'Resultado Esperado': 8.48528137423857}]
```

Para ampliar la información acerca de los casos de prueba, en el documento se ilustran los primeros tres casos, mostrando a qué situaciones corresponden. Observar las Figuras 1, 2 y 3.

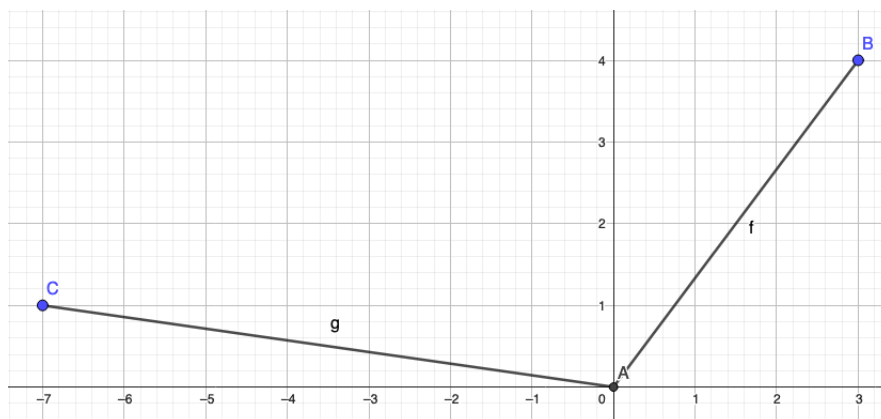


Figura 1: Graficado del primer caso de prueba, Entrada: (0,0,3,4,-7,1), Resultado Esperado: 5.0

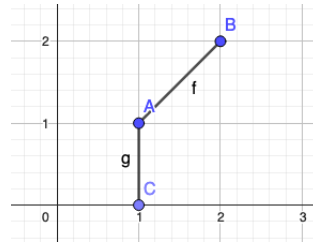


Figura 2: Graficado del segundo caso de prueba, Entrada: (1, 1, 2, 2, 1, 0), Resultado Esperado: 1.0

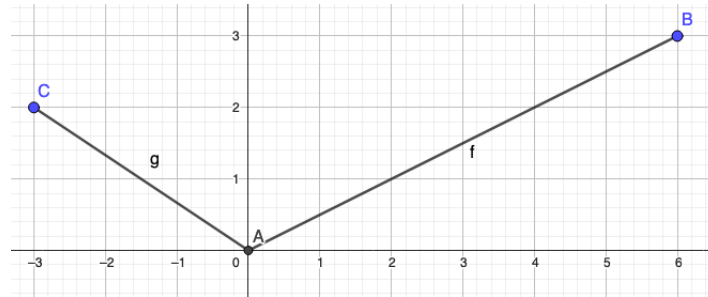


Figura 3: Graficado del tercer caso de prueba, Entrada: (0, 0, 6, 3, -3, 2), Resultado Esperado: 3.605551275463989

- Una empresa de logística cuenta con 2 tipos de camiones, sin embargo, las capacidades de estos en toneladas varían dependiendo del espacio que es necesario dejar libre para realizar recolección de paquetes durante la entrega de las demandas. Diseñar una función que reciba las 2 capacidades, que como se indican van a ser variables, además de la demanda. Esta función debe devolver entonces primero el número de camiones que se requieren de mayor capacidad para cubrir el total o la mayoría de la demanda, luego el número de camiones que se requieren del tipo con la menor capacidad para cubrir el excedente. También se debe producir un tercer valor, que indique si queda un excedente que no ocupe en su totalidad alguno de los camiones de menor capacidad que quedará pendiente. Este último valor será cero si no queda carga pendiente por distribuir. A continuación se presentan los casos de prueba, prestar mucha atención, porque la mayor capacidad, así como la menor, puede ir en cualquiera de las dos primeras entradas, además, la tercera entrada corresponde a la demanda. Finalmente, recordar que la salida tiene 3 valores en el siguiente orden estricto: número de camiones de capacidad mayor, número de camiones de capacidad menor y excedente que no fue acomodado porque no logran ocupar completamente alguno de los dos tipos de camión.

```
[{'Entrada': (10, 50, 235), 'Resultado Esperado': (4, 3, 5)},
{'Entrada': (100, 10, 200), 'Resultado Esperado': (2, 0, 0)},
{'Entrada': (40, 10, 440), 'Resultado Esperado': (11, 0, 0)},
{'Entrada': (10, 40, 440), 'Resultado Esperado': (11, 0, 0)},
{'Entrada': (2, 1, 23), 'Resultado Esperado': (11, 1, 0)},
{'Entrada': (7, 7, 1000), 'Resultado Esperado': (142, 0, 6)}]
```

- Desarrollar una función que reciba cantidades mayores o iguales a 1000 pesos, y retorne la cantidad de billetes mínima necesaria de 5000 y 1000 pesos para representar esa cantidad en papel moneda. Observar los casos de prueba y realizar el seguimiento para implementar con precisión el requerimiento.

Detalle de Casos:

```
[{'Entrada': 6000, 'Resultado Esperado': (1, 1)},
{'Entrada': 10000, 'Resultado Esperado': (2, 0)},
{'Entrada': 73000, 'Resultado Esperado': (14, 3)},
{'Entrada': 1000, 'Resultado Esperado': (0, 1)}]
```