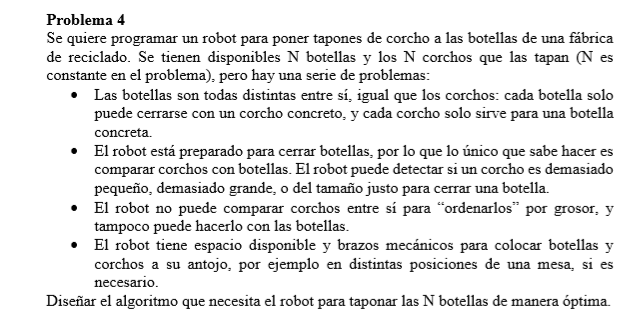
PL3

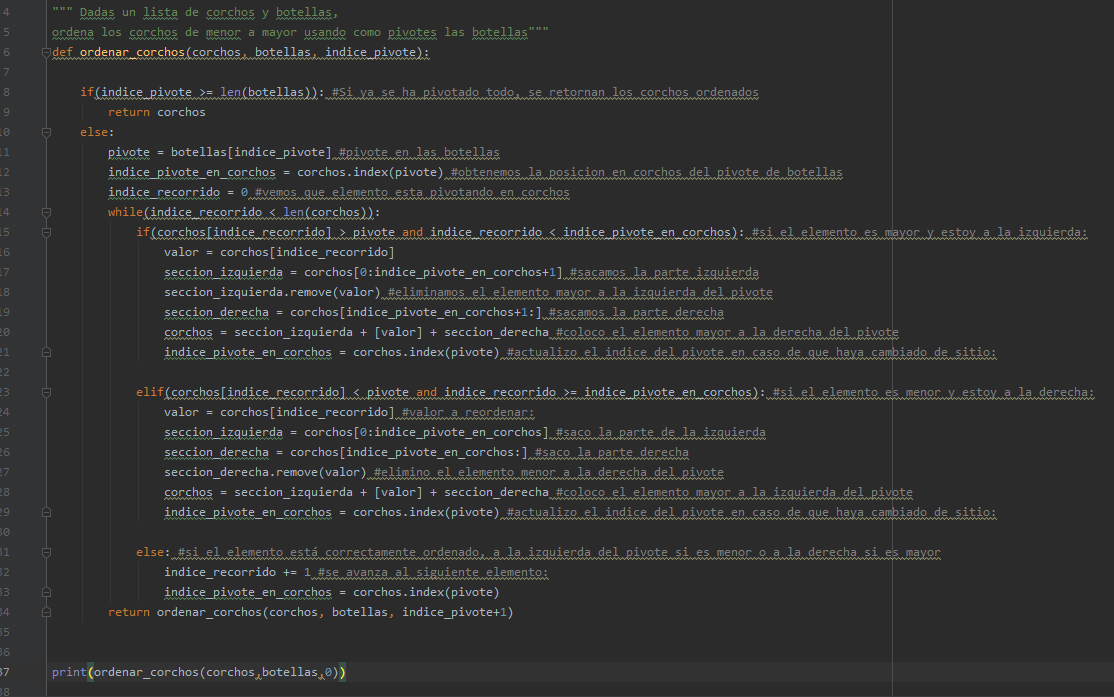
JAVIER MARTÍN GÓMEZ

47231977M

Este ejercicio se debe desarrollar de manera óptima, por lo que no valdría ordenar las dos listas, ya que el coste sería bastante grande. Para que el coste sea lo más bajo posible hay que desarrollar un algoritmo parecido al Quicksort, pero utilizando dos listas.

El algoritmo consiste en ir ordenando la lista de corchos a partir de los elementos de la lista botellas. Primero de todo, cogemos de pivote la primera posición de la lista de botellas, buscamos su posición en lista corchos y trabajamos con ese valor de pivote colocando los menores a su izquierda y los mayores a su derecha. Por ejemplo, si la lista de botellas es [2,3,7,1,3] y la de corchos [7,1,2,5,3], cogemos de pivote el 2, al ser el primero de la lista botellas. Una vez cogido el 2, buscamos la posición del 2 en la lista corchos y vamos comprobando si los mayores de 2 están a la derecha y los menores a la izquierda. Si no es así, se modifica la lista con los elementos bien colocados. Después se hará con el 3 y así sucesivamente hasta llegar a la última posición de la lista botellas que querrá decir que la lista corchos se ha ordenado correctamente de la manera óptima.

El algoritmo implementado en Python es el siguiente:

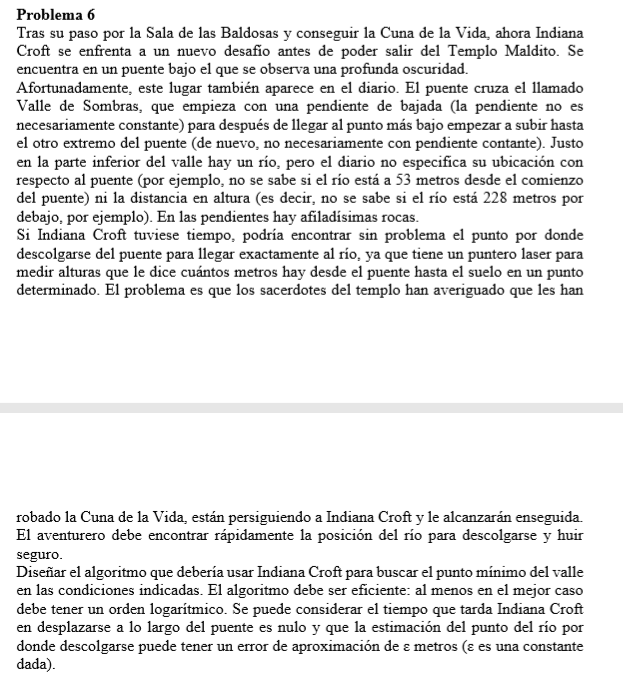


La salida del ejemplo comentado antes sería:



El coste de este algoritmo sería de O(n logn)

\*\*\*\*\*\*\*\*\*\*\*CORREGIDO\*\*\*\*\*\*\*\*\*\*\*



El ejercicio consiste en encontrar el punto donde el puente tenga la menor altura posible.

Antes de implementar el algoritmo hay que establecer una función que sería la del puente que atraviesa Indiana. A la función le damos un rango de valores que serán el inicio y el fin del puente. Seleccionamos 3 como la longitud máxima del puente.

Primero de todo obtenemos la longitud del puente (fin - inicio). Si la longitud es menor que 3, hallamos el valor mínimo del puente. SI no, vamos dividiendo el puente en 3 partes y nos vamos quedando con la que tenga valor mínimo y vamos llamando recursivamente a la función hasta que sea menor que 3 que nos retornará el valor mínimo.

Por ejemplo, para el caso de la función f(x)=x^2 con inicio=0 y fin=5 el resultado sería:

0,5

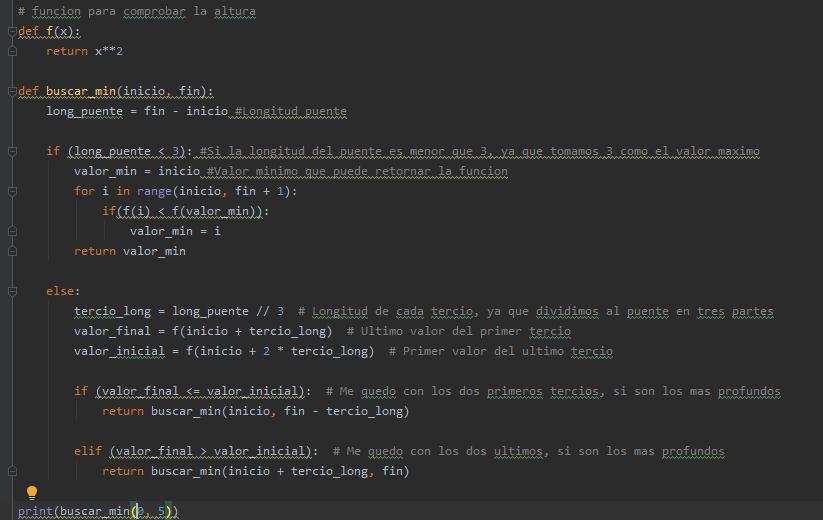
0,4, ya que el mínimo está en los dos primeros

0,3

0,2

0,1, como ya es menor que 3 devuelve el mínimo f(0)=0

El algoritmo implementado en Python es el siguiente:



El coste de este algoritmo sería de O(n logn)