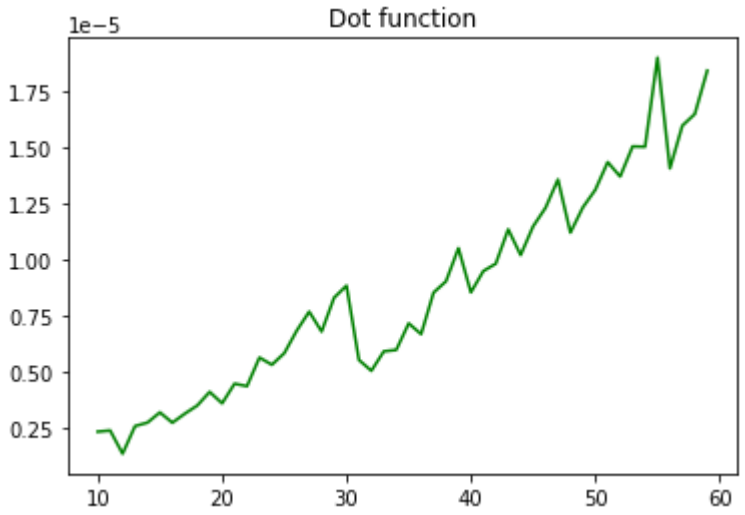
Nº pareja: 05
Nombres: Pablo Almarza y Carlos García

I-C. Cuestiones

1. ¿A qué función f se deberían ajustar los tiempos de multiplicación de la función de multiplicación de matrices? Usar el código inferior para ajustar valores de la forma a· f(t)+b a los tiempos de ejecución de la función que hayamos guardado en el array Numpy timings, para a continuación dibujar tanto los tiempos como el ajuste calculado por la función, y comentar los resultados.

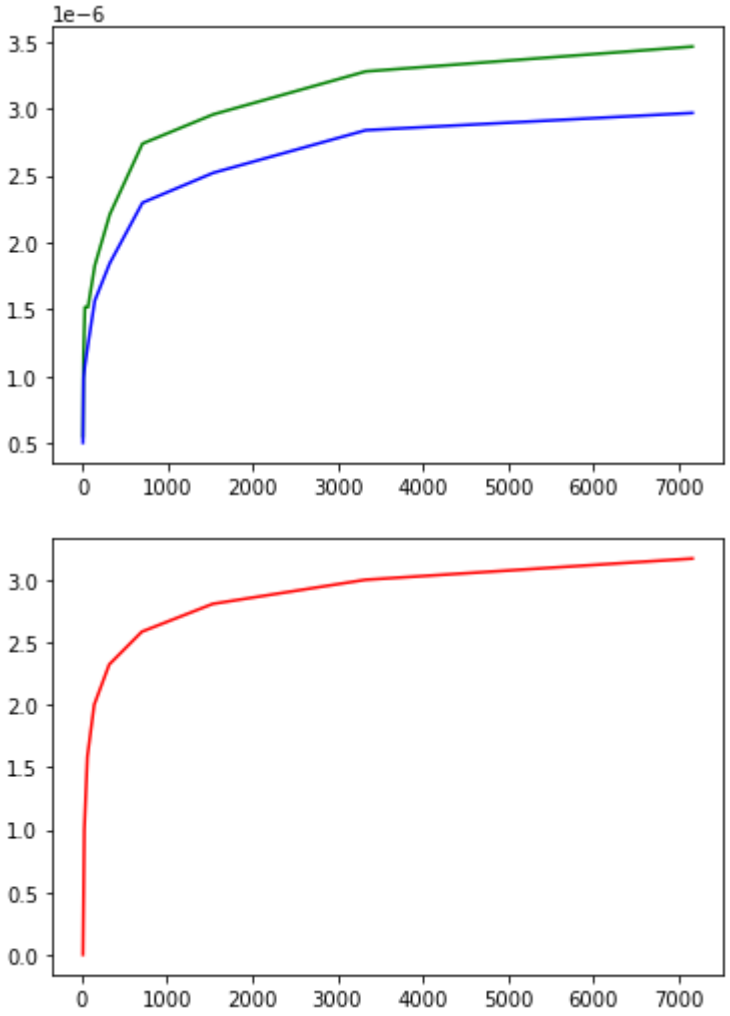
Function f will be a multiplication of the three dimensions of the two matrix, therefore f = a*b*c, where “a” is the number of rows of the first matrix, “b” is the number of columns of the second matrix and “c” is either the number of columns of the first matrix or the number of rows of the second matrix, since they are the same number. In our case, where we both matrix are square matrix, f = O(N³).

2. Calcular los tiempos de ejecución que se obtendrían usando la multiplicación de matrices a.dot(b) de Numpy y compararlos con los anteriores.



We can see in the files provided by both functions that the dot functions takes less time than our multiplication.

3. Comparar los tiempos de ejecución de las versiones recursiva e iterativa de la búsqueda binaria en su caso más costoso y dibujarlos para unos tamaños de tabla adecuados. ¿Se puede encontrar alguna relación entre ellos?

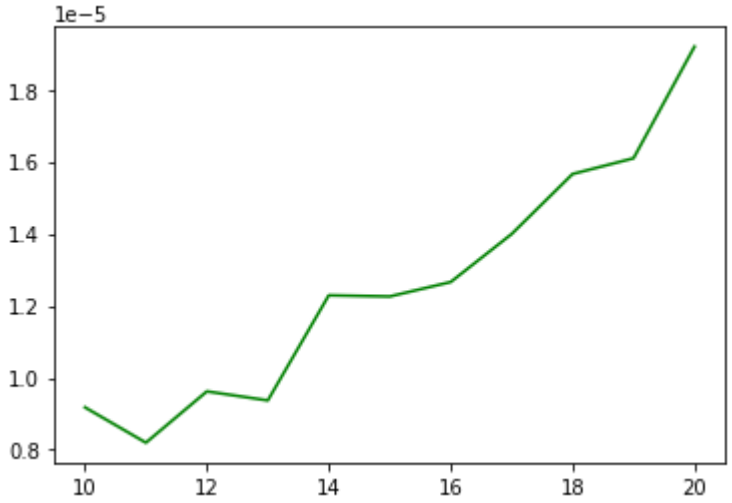


Green: recursive version.
Blue: Iterative version.
Red: O(log(n)) function.

We can find that both algorithms have the same shape, that corresponds to a function of O(log(n)).

II-D. Cuestiones

1. Analizar visualmente los tiempos de ejecución de nuestra función de creación de min heaps. ¿A qué función f se deberían ajustar dichos tiempos?



The function should adjust to a function of type O(n), which can be seen in this plot.

2. Expresar en función de k y del tamaño del array cual debería ser el coste de nuestra función para el problema de selección

   The function cost will be O(n*log(k)), being n the size of the array, since we perform heapify (that has a cost of log(n)) with a threshold k, and we perform it n times.

3. Una ventaja de nuestra solución al problema de selección es que también nos da los primeros k elementos de una ordenación del array. Explicar por qué esto es así y cómo se obtendrán estos k elementos.

   Since we create a min heap (with negative numbers), we get an ordered array, so we would get the k elements asked from the function.

4. La forma habitual de obtener los dos menores elementos de un array es mediante un doble for donde primero se encuentra el menor elemento y luego el menor de la tabla restante. ¿Se podrían obtener esos dos elementos con un único for sobre el array? ¿Como?

   It can be done with only one for, we use a heap of size 2, so we perform hepify for each element in the array so at the end of the array we have the 2 elements that we want.