Analisis y Diseño de Algoritmos

Marco Antonio Bastida Flores

13 Junio 2023

1 Quick Sort Algorithm

```
import random
import timeit

def partition(array, lo_index, hi_index):
    r = random.randint(lo_index, hi_index)
    array[r], array[hi_index] = array[hi_index], array[r]
    pivot = array[hi_index]
    i = lo_index
    for element in range(lo_index, hi_index):
        if array[element] < pivot:
            array[element], array[i] = array[i], array[element]
            i = i + 1
    array[hi_index], array[i] = array[i], array[hi_index]
    return i

def quick_sort(array, lo_index, hi_index):
    if lo_index < hi_index:
        pivot = partition(array, lo_index, hi_index)
        quick_sort(array, lo_index, pivot - 1)
        quick_sort(array, pivot + 1, hi_index)

def permutation(array):
    copy_array = array[:]
    length = len(array)
    for i in range(length):
        index = random.randint(0, length - 1)
        copy_array[i], copy_array[index] = copy_array[index], copy_array[i]
    return copy_array

def quick_sort_time():
    SETUP_CODE = '''
from __main__ import quick_sort
from __main__ import permutation
from __main__ import partition
mylist = [x for x in range(2000)]
my_copy_list = permutation(mylist)
'''
    TEST_CODE = '''
quick_sort(my_copy_list, 0, len(my_copy_list) - 1)
```

```
'''

    times = timeit.repeat(setup=SETUP_CODE,
                          stmt = TEST_CODE,
                          repeat=50,
                          number=1)
    total_time = 0
    for time in times:
        total_time += time
    print("2000: ",total_time/50)

if __name__ == "__main__":
    quick_sort_time()
```

2 Resultados

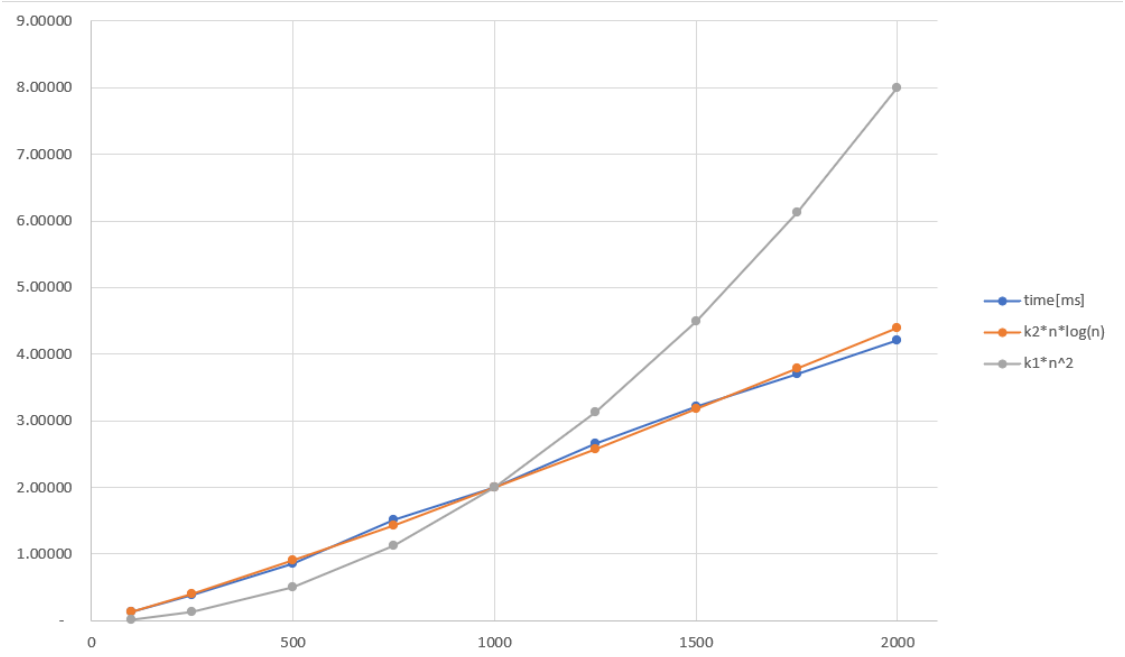

Figure 1: Grafica con resultados y funciones sobrepuestas

En la figura 1 se muestran que según los resultados de las corridas obtenidas, se aproxima más a un algoritmo $n * log(n)$.

Está manera de analizar algoritmos me pareció bastante interesante, ya que se puede realizar de una manera practica y resulta de una mejor manera mostrar los resultados con una gráfica y así ver más fácil en forma de graficas la comparación de las corridas.


View the Code on GitHub
Click to browse repository