

CAIM Lab, Session 8:

Locality Sensitive Hashing

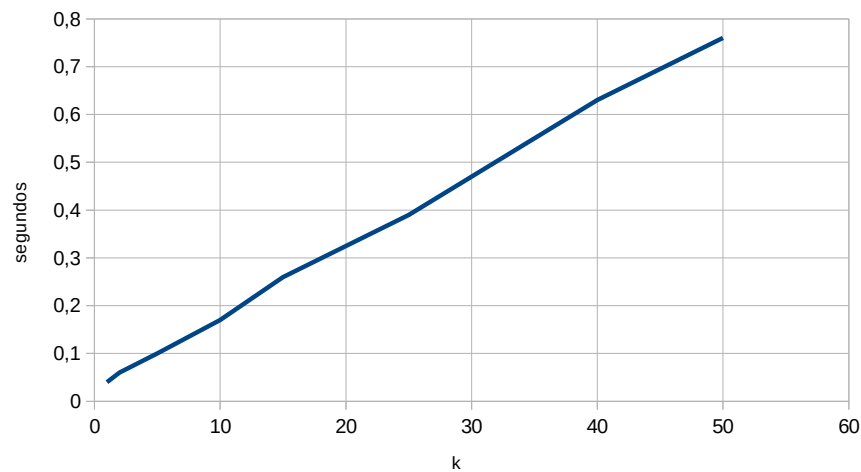
David Sànchez Peña

Tarea 1

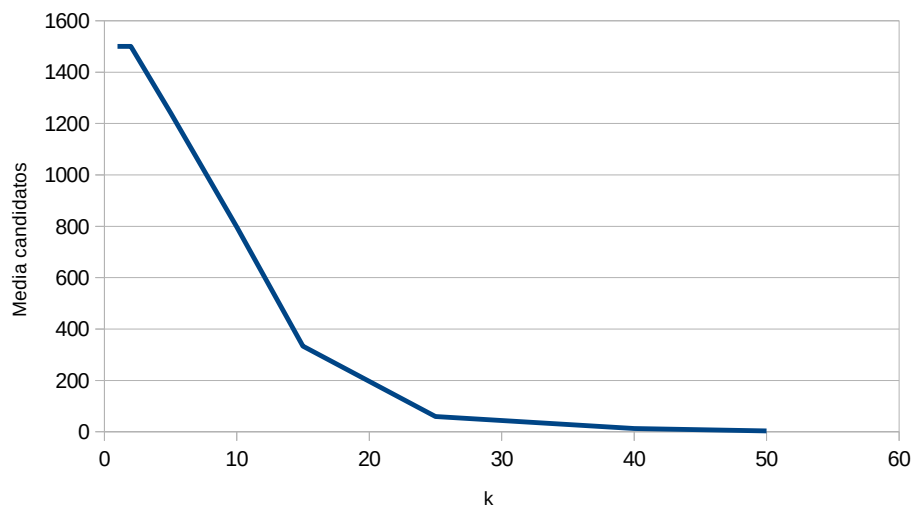
Parámetro k:

Para nuestro primer experimento fijaremos el valor $m = 5$. A medida que aumentamos el valor de k , aumenta el tiempo de ejecución de forma lineal:

k	1	2	5	10	15	25	40	50
segundos	0,04	0,06	0,1	0,17	0,26	0,39	0,63	0,76



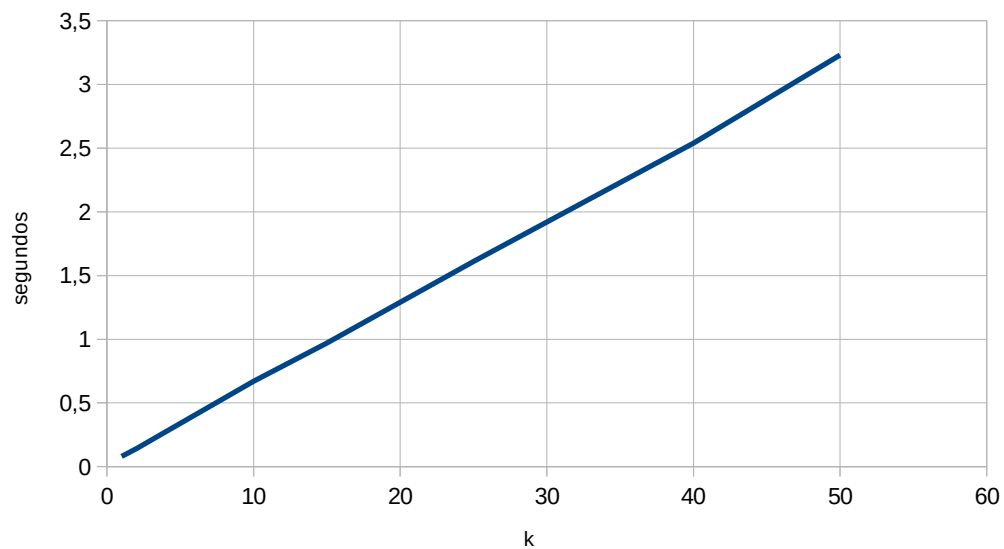
En lo que respecta al número de candidatos, a medida que aumentamos k , el número de candidatos se va reduciendo. Llegando a un punto en que el número de candidatos es incluso menor que 0:



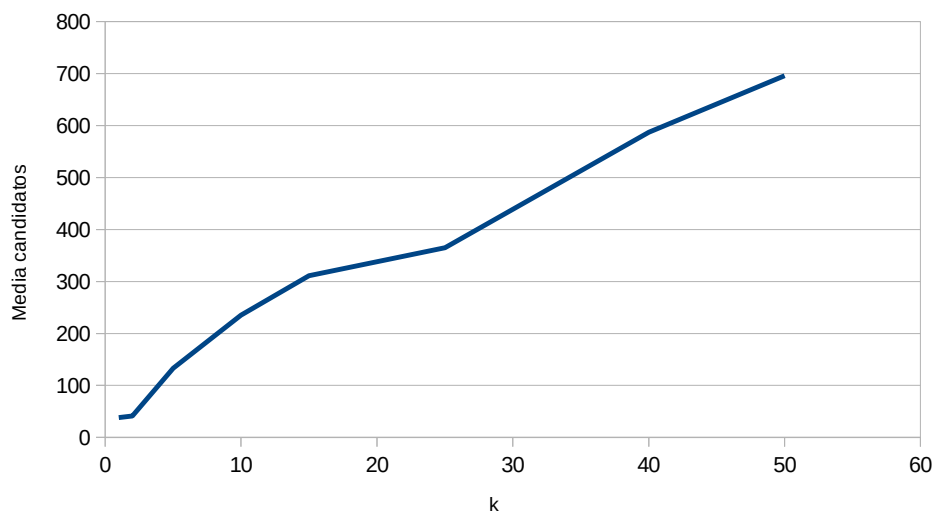
Parámetro m

Para el segundo experimento, fijaremos $k = 20$. Entonces, a medida que vamos aumentando m , vemos un efecto similar al del anterior experimento, donde el tiempo de ejecución aumenta de forma lineal.

k	1	2	5	10	15	25	40	50
segundos	0,08	0,14	0,34	0,67	0,97	1,61	2,54	3,23



Pero esta vez vemos el efecto contrario que en el experimento anterior, ya que a medida que m aumenta, también lo hace el número de candidatos:



Tarea 2

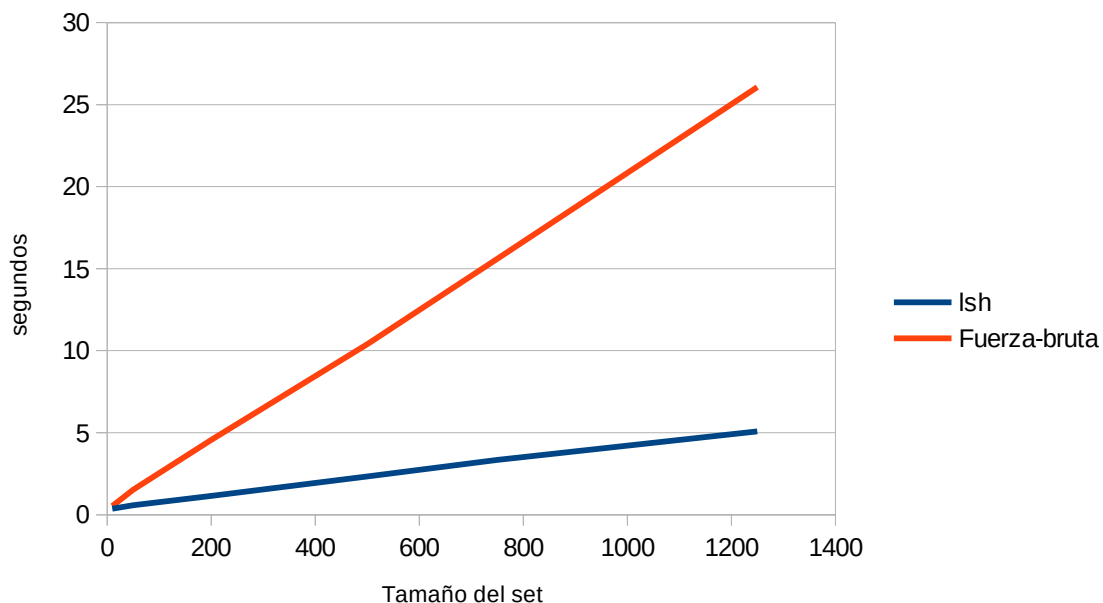
Antes de seguir realizando experimentos, se nos pide que implementemos algunas funciones en python.

Primero de todo tenemos que hacer una función que compute la distancia entre dos imágenes de entrada. Para ello utilizaremos la distancia de Hamming.

Después, una función que dada una imagen, encuentre la imagen más cercana utilizando un algoritmo de fuerza bruta.

Y finalmente, implementaremos un método de búsqueda basado en hashing, que tendrá un funcionamiento parecido al de fuerza bruta, pero solamente iterará sobre el set de candidatos.

Para empezar, veremos como afecta el tamaño del set de imagenes en la ejecución del programa. Comparando el tiempo de ejecución de estos dos métodos, obtenemos la siguiente gráfica:



Después podemos observar que si el valor de m es demasiado pequeño, los resultados de los dos métodos son distintos. Lo mismo pasa cuando aumentamos el valor de k . Esto es debido a que al alterar estos valores, el número de candidatos se reduce, y entonces aumenta la dificultad de encontrar la imagen óptima.