INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA CAMPUS TLAXCALA

INGENIERÍA EN INTELIGENCIA ARTIFICIAL

ANÁLISIS Y DISEÑO DE ALGORITMOS

DETERMINACIÓN DEL UMBRAL PARA EL PROBLEMA DE KNN

ALUMNOS: ANGEL MIGUEL SÁNCHEZ PÉREZ
SAMUEL SORIANO CHÁVEZ
GUILLERMO CARRETO SÁNCHEZ

FECHA DE ENTREGA: 26 DE ABRIL DE 2024

INTRODUCCIÓN.

Para la resolución de problemas se parte normalmente desde la solución más simple aunque no siempre es la más eficaz debido al tamaño del problema y a posibles resoluciones no tan obvias que se derivan de análisis detallados y de creatividad personal o colectiva, de manera usual el método de solución más simple aplica fuerza bruta al probar todas las posibilidades o caminos a los que nos puede llevar un problema, es el más sencillo pues no necesita de un razonamiento complejo y sólo utiliza una secuencia de pasos sencillos, pero este tipo de solución deja de ser viable cuando el tamaño del problema original es demasiado grande y requiere de una gran cantidad de recursos para desarrollarse por la forma más simple, es por eso que es necesario pensar más allá de lo convencional, buscando procesos que eviten realizar todas las posibles combinaciones derivables, y es de esta idea que se piensa en un método para dividir el problema y pensar en subpartes de un conjunto, para solucionar cada una de estos componentes del total, para después, combinar todas las posibles soluciones, este método de resolución adquiere el nombre de “divide y vencerás” y es útil para desafíos

con grandes cantidades de datos, este proceso se conforma de tres partes:

1° Dividir: Divide el problema grande en subproblemas más pequeños y manejables. Esta división debe ser exhaustiva y garantizar que cada subproblema sea más fácil de resolver que el problema original.

2°Conquistar: Resuelve cada subproblema de forma independiente y recursiva. Si los subproblemas son lo suficientemente pequeños, resuélvelos directamente.

3°Combinar: Combina las soluciones de los subproblemas para obtener la solución del problema original. Esta etapa es crucial para garantizar que la solución global sea correcta.

Habiendo definido ambos tipos de algoritmos (“fuerza bruta” y “divide y vencerás”) se necesita profundizar en la teoría para determinar el punto en donde uno de estos enfoques se vuelve más eficiente que el otro para resolver un problema en específico y este punto es llamado “umbral” y es crucial para elegir el enfoque adecuado para resolver un problema dado.

En términos generales:

- ❖ Divide y vencerás: Este enfoque tiende a ser más eficiente cuando el tamaño del problema es lo suficientemente grande como para dividirlo en subproblemas más pequeños y manejables. Si el problema puede ser dividido en partes que pueden ser resueltas independientemente y luego combinadas, entonces el enfoque de "divide y vencerás" puede ser preferible.
- ❖ Fuerza bruta: Este enfoque, por otro lado, es más adecuado cuando el tamaño del problema es pequeño y la búsqueda exhaustiva de todas las posibles soluciones no consume demasiados recursos computacionales. Cuando el tamaño del problema es pequeño y no hay una estructura clara para dividirlo en subproblemas más pequeños, la fuerza bruta puede ser la mejor opción.

El umbral depende de varios factores, como el tamaño del problema, la eficiencia de los algoritmos utilizados, los recursos computacionales disponibles y las restricciones de tiempo. En la práctica, se pueden realizar análisis teóricos y pruebas empíricas para determinar cuál enfoque es más adecuado para un problema específico en función de estos factores.

El proceso para obtener el umbral es el siguiente:

- ➢ Caracterización del problema: Comprender completamente el problema y sus características es el primer paso. Esto incluye el tamaño del problema (por ejemplo, el número de elementos a procesar), la estructura del problema (si es posible dividirlo en subproblemas más pequeños), y los recursos computacionales disponibles.

- Complejidad algorítmica: Analizar la complejidad temporal y espacial de ambos enfoques para el problema en cuestión. Para "divide y vencerás", esto implica determinar el número de operaciones requeridas para dividir el problema, resolver los subproblemas y combinar las soluciones. Para la fuerza bruta, implica determinar el número de combinaciones o iteraciones necesarias para evaluar todas las posibles soluciones.
- Comparación de complejidades: Comparar las complejidades temporales de ambos enfoques en función del tamaño del problema. Identificar el punto en el que una estrategia se vuelve más eficiente que la otra.
- Análisis experimental: Realizar pruebas empíricas con conjuntos de datos de diferentes tamaños para validar los resultados teóricos. Esto implica ejecutar los algoritmos en diferentes instancias del problema y medir el tiempo de ejecución o el consumo de recursos. Estas pruebas pueden ayudar a confirmar el umbral identificado en el análisis teórico.
- Refinamiento y ajuste: Ajustar el análisis teórico y experimental según sea necesario para reflejar con mayor precisión el desempeño de los algoritmos en condiciones reales.

Mediante estos pasos, se puede obtener un umbral que indique cuándo es más eficiente utilizar el enfoque de "divide y vencerás" en comparación con la fuerza bruta para un problema específico. Este umbral puede variar según las características del problema y los recursos disponibles, por lo que es importante realizar un análisis exhaustivo y validar los resultados mediante pruebas experimentales.

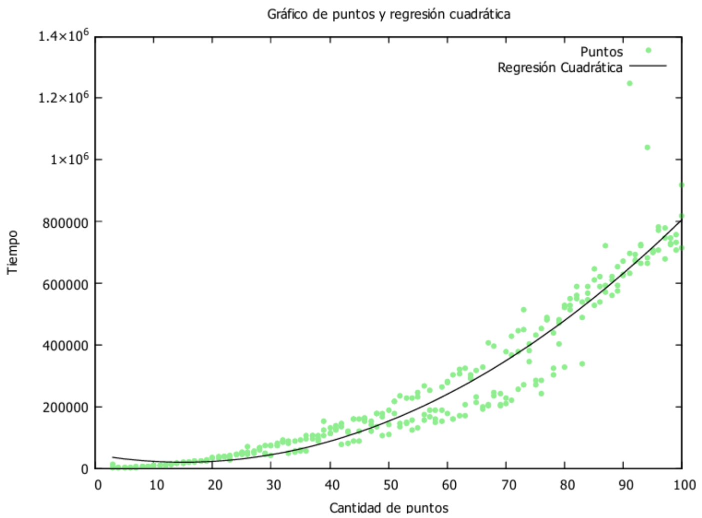
DESARROLLO.

La implementación tanto del algoritmo de fuerza bruta y divide-vencerás de los puntos más cercanos fue lo primero que se desarrolló. A partir de la implementación de estos algoritmos en el lenguaje c, se empezó a ejecutar diferentes n números de puntos para obtener el tiempo en nanosegundos.

El algoritmo implementado resultante se anexa en esta tarea. En él se encuentra de la misma forma, las dos funciones que ocupamos para obtener las regresiones de ambos algoritmos, la lineal para el algoritmo divide y vencerás, y la cuadrática para el algoritmo de fuerza bruta. También, dentro del código se implementa los comandos del script de gnu-plot para poder graficar nuestros resultados.

Una vez generado los n cantidad de puntos y sus tiempos, fuimos observando cómo es que se comportan estos algoritmos bajo ciertas condiciones. Para no tardar tanto en la práctica, automatizamos el proceso para generar tiempos, con dos ciclos for. Vimos que, para las regresiones, afecta bastante el aumento de los n cantidad de puntos, ya que si la serie iba aumentando de 1 en 1 la regresión resultante era totalmente diferente a una serie con paso 5 en 5.

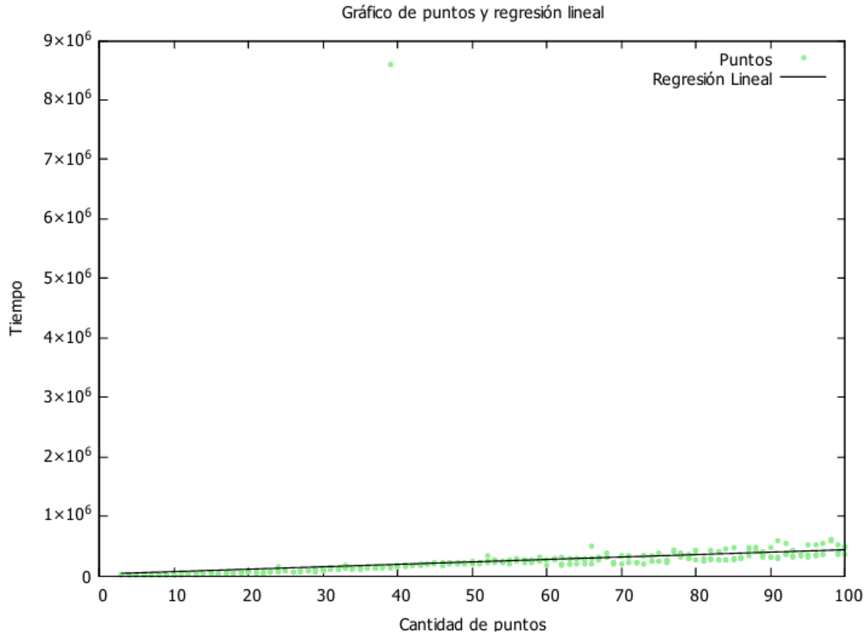
Entonces, los resultados para ambos algoritmos son los siguientes:



$$h(n) = 109.035443n^2 - 3279.189888n + 44172.867864$$

Este gráfico representa el comportamiento del algoritmo fuerza bruta, en él se observa cómo es que efectivamente es un algoritmo con complejidad n^2, la curva que se desarrolla pertenece a un comportamiento de complejidad cuadrática, y los puntos verdes representan las coordenadas dadas por la cantidad de puntos para el eje x y los tiempos obtenidos en el proceso de medición.

Para el algoritmo divide y vencerás, la regresión lineal genera una función que tiene comportamiento lineal. Como se muestra en la siguiente gráfica:



$$g(n) = 4093.75n + 28521.86$$

Algo que notamos, es que su comportamiento es muy constante, no hubo puntos que salieran bastante del trazado de la función, a comparación del algoritmo fuerza bruta.

Lo siguiente que se hizo fue, proceder a obtener nuestro umbral o nuestro n0. Para ello, una vez que logramos obtener regresiones con coeficientes positivos en la variable n con mayor grado y de la constante (notamos que cuando estos eran positivos, el n0 era un valor real y no uno imaginario), utilizamos el método teórico para la obtención de n0.
Entonces, definiendo la función h(n) que corresponde a la regresión cuadrática del algoritmo fuerza bruta, y también la función g(n) que se obtiene de la regresión lineal del algoritmo divide-vencerás.

$$h(n) = 109.035443n^2 - 3279.189888n + 44172.867864$$

$$g(n) = 4093.75n + 28521.86$$

Sabemos que el algoritmo de divide-vencerás se basa precisamente en la recursión, donde se divide el algoritmo en dos subproblemas y que la entrada n se está dividiendo en dos constantemente, de ahí se obtiene el valor de a=2 y b=2 para la fórmula:

$$h(n) = ah\left(\frac{n}{b}\right) + g(n)$$

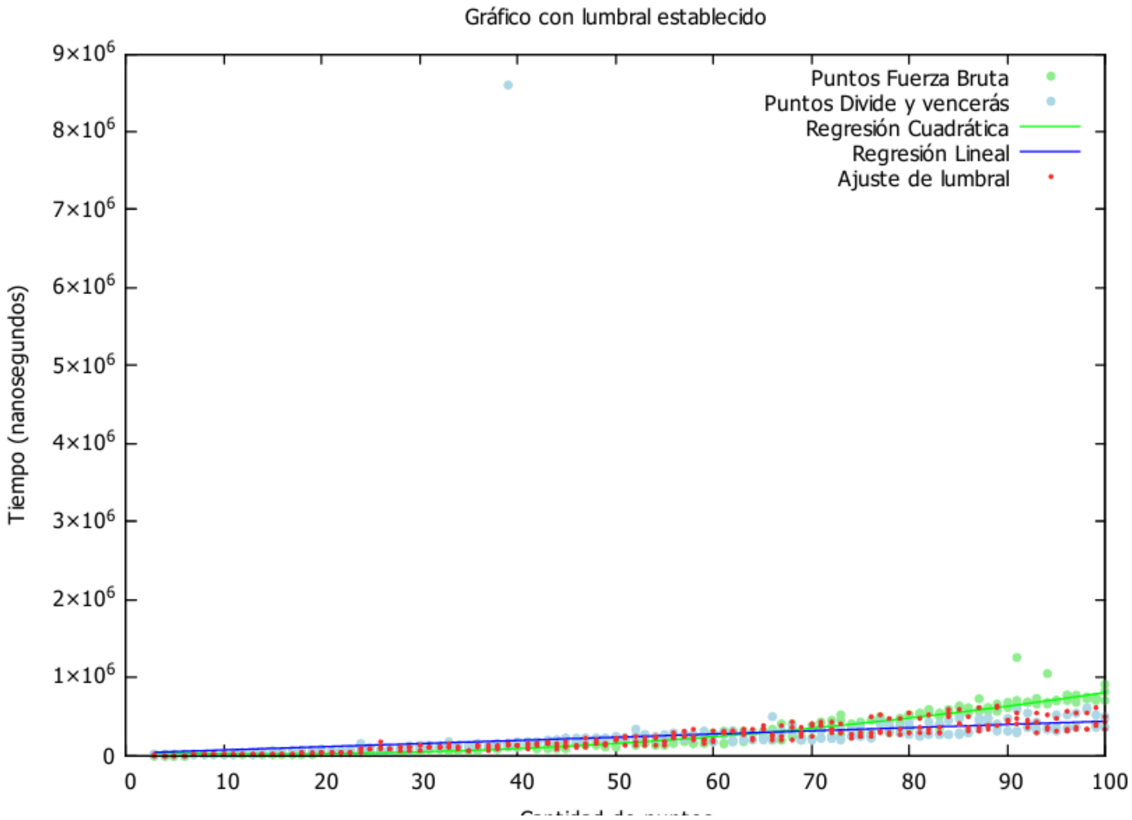
Sustituyendo:

$$109.035443n^2 - 3279.189888n + 44172.867864 = 2((109.035443 * \frac{n^2}{4}) - (3279.189888 * \frac{n}{2}) + 44172.867864)$$

Utilizando una calculadora, se obtiene que las raíces de la ecuación son:

$$n0_1 = 89.9192$$
$$n0_2 = -14.82901$$

De las cuales, solo nos interesa tomar la raíz positiva, n0=89.91. Una vez con nuestro lumbral, procedimos hacer la prueba correspondiente con el lumbral adecuado. Los resultados fueron los siguientes.



En la gráfica se observa que los puntos rojos definen el comportamiento de los algoritmos combinados, y que ajustar el lumbral hace que el algoritmo obtenga el máximo provecho de los otros dos algoritmos, el de fuerza bruta y el de divide-vencerás. El hecho de que el algoritmo ajustado y el de divide-vencerás se comporten de manera lineal, no permite que se muestre tanto la curva que genera el de fuerza bruta.

CONCLUSIÓN.

Después de realizar la actividad, nos pudimos dar cuenta de muchas cosas importantes. La primera fue que hay diferentes formas de implementar un mismo problema y esto no significa que los algoritmos sean igual de eficientes si se usa un método u otro, la efectividad depende de lo que uses para implementar el código que resuelve el problema. Luego, al hacer las pruebas con los dos métodos de implementación del problema, nos dimos cuenta de que ambos funcionan de manera muy eficiente cuando se tienen pocos datos de entrada, pero hay un momento en el que la implementación de fuerza bruta empieza a empeorar de manera muy significativa volviendo así su gráfica a la forma cuadrática. Finalmente nos percatamos de que la implementación de fuerza bruta no siempre es peor que la de divide y vencerás, cuando los datos son pocos, la implementación de fuerza bruta puede llegar a ser más efectiva que la de divide y vencerás, es por eso que después de hacer las pruebas, obtuvimos las regresiones cuadrática y lineal respectivamente de las implementaciones con el fin de encontrar una cantidad de datos para la que es conveniente usar fuerza bruta. Al alcanzar esa cantidad de datos, es recomendable usar divide y vencerás para así hacer que la combinación de ambos métodos forme un nuevo umbral que sea más efectivo o igual de efectivo que el divide y vencerás. Como se pudo observar en nuestros resultados, el nuevo umbral es muy similar al comportamiento de la regresión lineal de divide y vencerás y se obtiene el máximo provecho del algoritmo.

REFERENCIAS.

(de Algoritmos, s/f)
de Algoritmos, A. y. D. (s/f). Algoritmos “Divide y Vencerás”. Ugr.es. Recuperado el 26 de abril de 2024, de https://elvex.ugr.es/decsai/algorithms/slides/3%20DV.pdf

(Impera, s/f)
Impera, D. et. (s/f). El esquema “Divide y vencerás” 1. Cartagena99.com. Recuperado el 26 de abril de 2024, de https://www.cartagena99.com/recursos/alumnos/apuntes/Capitulo5.pdf

(Mallen, s/f)
Mallen, Y. O. (s/f). Divide y vencer´as. Cartagena99.com. Recuperado el 26 de abril de 2024, de https://www.cartagena99.com/recursos/alumnos/apuntes/210628121342-trans-divideyvenceras.pdf