



UNIVERSIDAD DEL NORTE

DEPARTAMENTO DE INGENIERIAS DE SISTEMAS Y COMPUTACION

Asignatura: Algoritmo y complejidad Docente: Misael Diaz Maldonado

I+D Report

Nilson David Diaz Hasbun: 200152551

Santiago Hernández Cadavid: 200153162

{Hasbunn, santiagoacadavid} @uninorte.edu.co

IST 3264

Resumen. A lo largo del curso hemos analizado la complejidad temporal de varios algoritmos y como está relacionado con el número de iteración que realiza, para este proyecto I+D se realizara un análisis de la complejidad temporal del algoritmo de fuerza bruta implementado las (LinkedList) frente el uso de ArrayList como se realizó en el laboratorio anterior. El programa genera aleatoriamente una serie de coordenadas de tamaño N respecto al eje X.

II. INTRODUCCION

Para empezar con este proyecto estudiamos la complejidad temporal del algoritmo Closest Pair utilizando LinkedList para lograr obtener una comparación con el laboratorio anterior donde utilizamos ArrayList para demostrar cómo tiene un comportamiento menos eficiente utilizando LinkedList para encontrar la menor distancia entre N puntos.

III. METODOS Y MATERIALES

Para desarrollar este proyecto creamos un conjunto de coordenadas X, Y de forma aleatoria y luego se almacenan en una lista que se convierte en un LinkedList para su uso, las coordenadas con respecto al eje X. A sí mismo en la subrutina recursiva BruteForce que utiliza la lista previamente ordenada para encontrar el par más cercano. Para así conocer el cómputo de dicho algoritmo se almacenan los datos en el fichero .txt. Esto se hace tomando un conjunto de coordenadas de tamaño 50 hasta un tamaño de 20000 con un incremento de 3/2

IV. CODIGO

Algoritmo 1 de Fuerza Bruta

```
dmin ← INF
for i = [1, N - 1] do
  for j = [i + 1, N - 1] do
    d ←
    distancia(coordenadas, i, j)
    if d < dmin then
      first ← i
      second ← j
      dmin ← d
    end if
  end for
end for
return(first, second, dmin)
```

V. RESULTADOS

Utilizamos el archivo .txt generado anteriormente que se utilizó para generar una gráfica con los tiempos respecto al tamaño N obtenidos anteriormente. Como pueden ver a continuación en la tabla con los tiempos de ejecución y el gráfico el algoritmo tiene un comportamiento cuadrático de $O(N^2)$ como se puede observar

N	Comparaciones	Tiempo
50	167	20277
75	283	22668
112	300	25683
168	526	51651
252	572	44980
378	1183	138062
567	2098	209469
850	2336	488402
1275	3709	1000762
1912	4878	2087189
2868	7385	4756316
4302	15501	10931055
6453	18502	25140868
9679	28667	62651051
14518	37132	136677333

Figure 1. Results

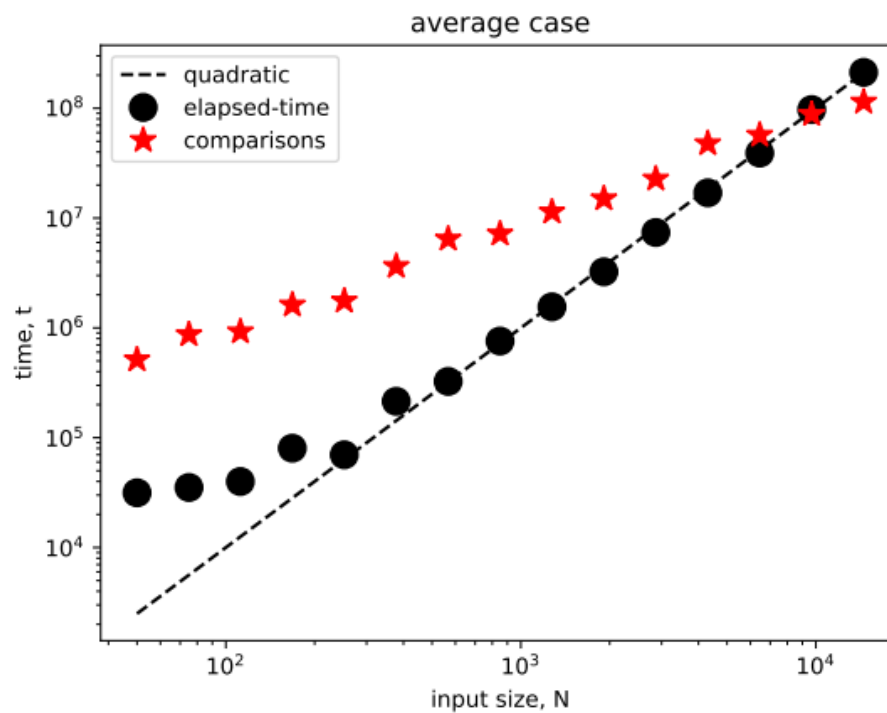


Figure 2. Algorithm Complexity

Como se puede ver en el gráfico, los datos tienen un comportamiento cuadrático, como era de esperar al utilizar LinkedList.

VI. CONCLUSION

Para concluir que la complejidad esperada se ha cumplido para este proyecto. como poder observar en los resultados el tiempo de ejecución y las comparaciones son proporcionales al tamaño de las n de las coordenadas, además la complejidad del algoritmo es $O(N^2)$ como se esperaba. Con estos resultados podemos comparar la complejidad del algoritmo anterior que era $O(N)$ en el que se utilizaba LinkedList como se pedía en este proyecto I+D. durante el desarrollo de este proyecto no hubo algún error ya que la mayor parte del algoritmo fue reutilizar lo que se hizo en el laboratorio anterior, haciendo unos breves cambios para que funcionara con LinkedList.