



UNIVERSIDAD DEL NORTE

DEPARTAMENTO DE INGENIERÍAS SISTEMAS Y COMPUTACION

Asignatura: Algoritmo y complejidad Docente: Misael Diaz Maldonado.

## Lab 3: Closest Pair

Nilson David Diaz Hasbun

{Hasbunn} @uninorte.edu.co

Octubre 2022

**Resumen.** En el siguiente informe se encuentra el Desarrollo del comportamiento de un algoritmo que compara las distancias de puntos generados aleatoriamente y determina que par de punto es el más cercano.

### 1. INTRODUCCIÓN

El objetivo principal de estas entregas es determinar la complejidad del algoritmo poniéndole diversas pruebas para elaborar una gráfica que nos permitirá visualizar los resultados obtenidos de los tiempos de ejecución de dicho algoritmo.

### 2. PROCEDIMIENTOS

Para el desarrollo de este laboratorio, el algoritmo de closest pair o par mas cercano es usado para hallar la mínima distancia entre partículas que se encuentran en un array con coordenadas x, y, primeramente estos dos vectores son creados por x[], y[] en un tamaño N, a estos se la asignan números aleatorios para cada posición en los vectores. Después de la creación de los vectores y la organización, se crea una matriz de tamaño  $n * N$  donde las coordenadas de X se almacenan en la primera fila Y en la segunda fila de la matriz. Finalmente, la matriz se divide en 2 nuevas matrices, en las que las coordenadas resultantes de la división de la matriz original.

### 4. ALGORITMO EMPLEADO

```
dmin <- I N F
```

```
Para i = [1, N-1] hacer
```

```
    Para J = [I+1, N+1] Hacer
```

```
        d<- Distancia(coord,I,J)
```

```
        if d < dmin entonces
```

```
            first<- I
```

```
            second<- J
```

```
dmin<-d
```

```
FinPara
```

```
FinPara
```

```
return(frist,second,dmin)
```

```
FinProceso
```

## 5. Resultados

1000	3909	157738
1500	4661	143313
2250	8311	253624
3375	11517	519339
5062	17152	775725
7593	28451	1111140
11389	36435	1122306
17083	65452	2112226
25624	82919	2532246
38436	135556	4326527
57654	207662	6935723
86481	283007	8928651
129721	509481	16238570
194581	604124	21066570
291871	1064268	44429146
437806	1504585	58938442
656709	2206530	89223998
985063	3729954	146647837
1477594	4694684	217160298

Fig.1 Resultados

## TimeComplexity

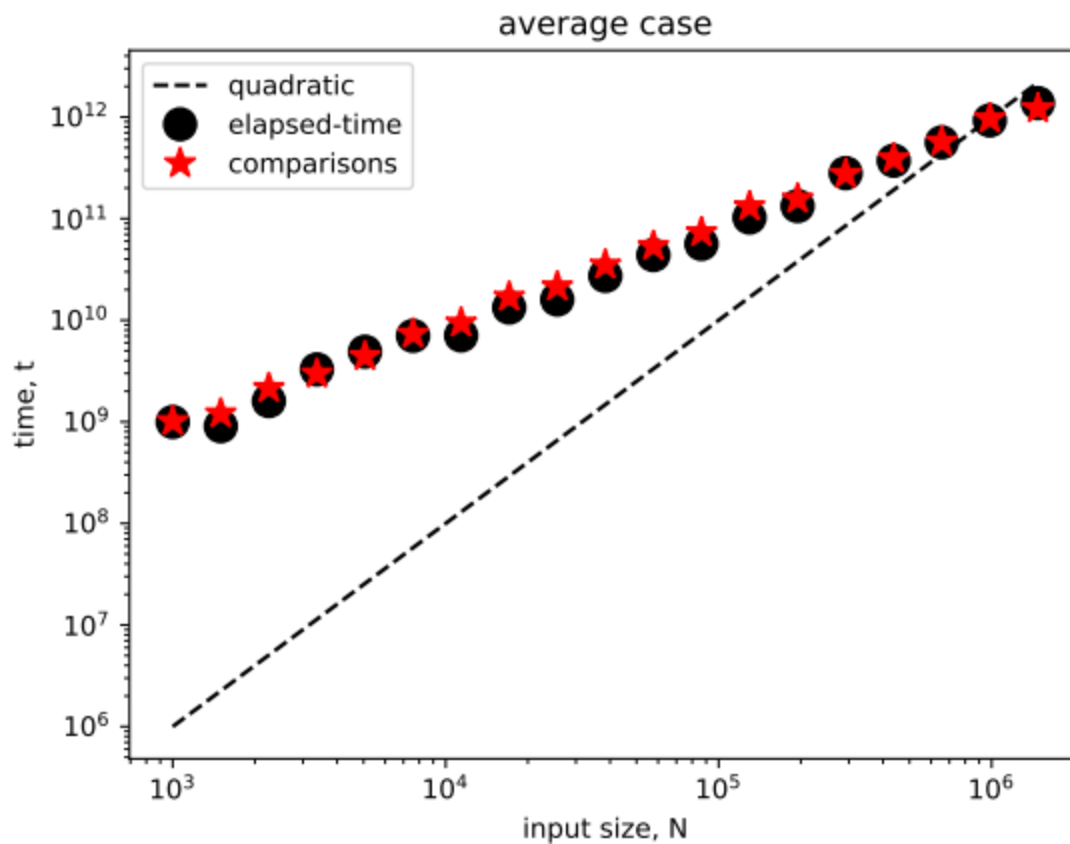


Fig2. Complejidad del algoritmo

## 5. CONCLUSIONES

Como podemos observar y comprobar que el algoritmo de fuerza bruta tiene un comportamiento lineal como se observó anteriormente en el gráfico generado, dándonos como resultado que el tiempo de ejecución es proporcional al tamaño de  $N$  coordenadas generadas en cada iteración, a su mismo pudimos encontrar la complejidad de dicho algoritmo que en este caso es  $O(n)$ , para añadir no se encontraron dificultades a la hora de realizar el laboratorio como a la hora de hacer las pruebas.