Closest Pair Experiment Analysis
Curso : Algoritmos y Complejidad
Código estudiantil: 200156522
Fecha : 25 de noviembre del 2022
Alumna : Natalia Martínez Díaz
Profesor: Misael Díaz Maldonado

Abstract

This document compares the behavior of two different programs implemented in java that generate n coordinates and determines among all the coordinates generated the pair of coordinates that have the smallest distance between them. The difference between them is that in one program Arraylist are used to store the n coordinates and the other uses a Linked List implemented by myself. The objective is to explain why one program is more efficient than the other. The program were Arraylist were used was already analyzed in the first lab. The second program will be analyzed and then it will be compared to the other program.

1 Introduction

In the world of programming they exist many different types of data structures to solve any problem. There are cases were it is more beneficial to use a certain type of structure than the other even though both work. The main factors for choosing the right data structure are: it solves the problem, it is efficient, and space in memory. In this analysis we will focuss on ArrayLists and Linked Lists for solving the closest pair problem.

2 Problem Definition

Linked lists are not used as much as before by Java programmers. Arraylist tend to be more efficient and constant for search opperations. But linked lists are better for data manipulation. For this problem, we require the most efficient way of find the closest pair. Using ArrayList was already tested and analyzed, now Linked Lists will be tested for comparison.

3 Methodology

Java Netbeans was the software used to the program in Java. The program is able to generate n amount of coordinates and sort the coordinates in ascending order depending on the x value.Now, instead of using ArrayList for storing the coordinates linked lists are used. For sorting the list we used bubble sort. The same two methods to find the closest pair are implemented for this program:

1. By Brute Force:
    - By brute force two while loops are used to compare every single coordinate with the other coordinates. The distance of the coordinates compared is calculated and to determine which is the closest pair there is a conditional inside the while loops.
2. By Divide and Conquer:
    - The function brute force is also part of the solution of the Divide and Conquer method. The difference is that by Divide and Conquer there is a an additional recursive function that separates the Linked List of coordinates into many sub Linked Lists. When the size of the Linked List is less or equal to 3, the brute force function is executed. Also, the purpose of this separation is to find the group that contains the smallest distance and with that distance create another sub linked list that only contains coordinates that are that distance away or smaller from the mid coordinate.

To save the amount of coordinates (n), amount of iterations, and time execution we used an Array for each mentioned. Also, the program also generates a file with the information collected after execution (Amount of numbers, amount of iterations, and time execution) to be able to create the graphs and tables shown on the results section. The graphs on the result section were generated by a Python Program developed by Misael Díaz [1].

4 Results

4.1 Program with ArrayLists

Brute Force:

N	Comparisons	Time execution (NanoSeconds)
8	79	1123909
16	283	86490
32	1113	220025
64	4365	642455
128	17086	1506843
256	67217	1301010
512	266071	5711350
1024	1057478	11931927
2048	4214252	51688149
4096	16821190	199356279
8192	67205113	678447227
16384	268644271	2686747416
32768	1074192336	10690572578
65536	4295934201	51441695018
131072	17181934043	305260394975
262144	68723869004	1660029726918

Table 1: Brute Force Results

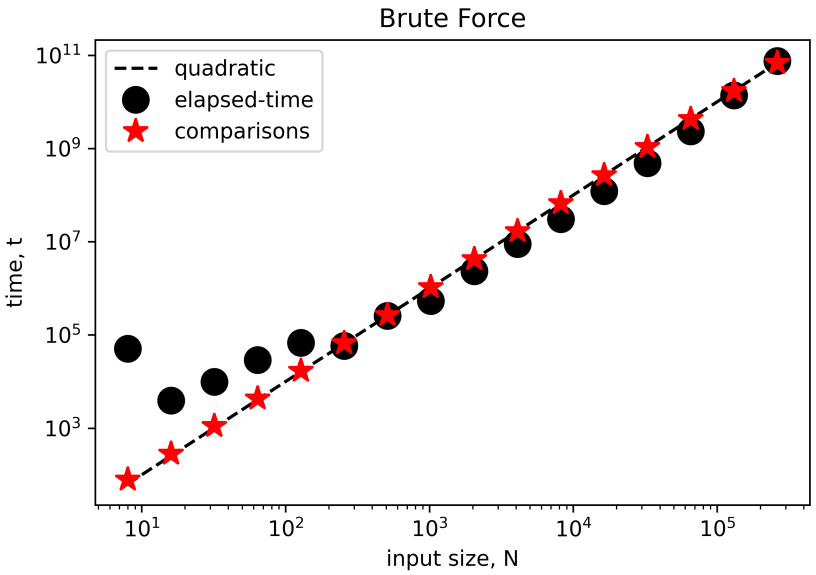

Figure 1: Brute Force Graph

Divide and Conquer:

N	Comparisons	Time execution (NanoSeconds)
8	44	1046945
16	72	94924
32	127	270067
64	258	546082
128	539	616102
256	1088	793045
512	2204	1168157
1024	4482	2280777
2048	8944	5477303
4096	17890	9259060
8192	35738	10542270
16384	71479	16714209
32768	142807	24026676
65536	286380	47108325
131072	573244	120369049
262144	1147391	347920057

Table 2: Divide and Conquer Results

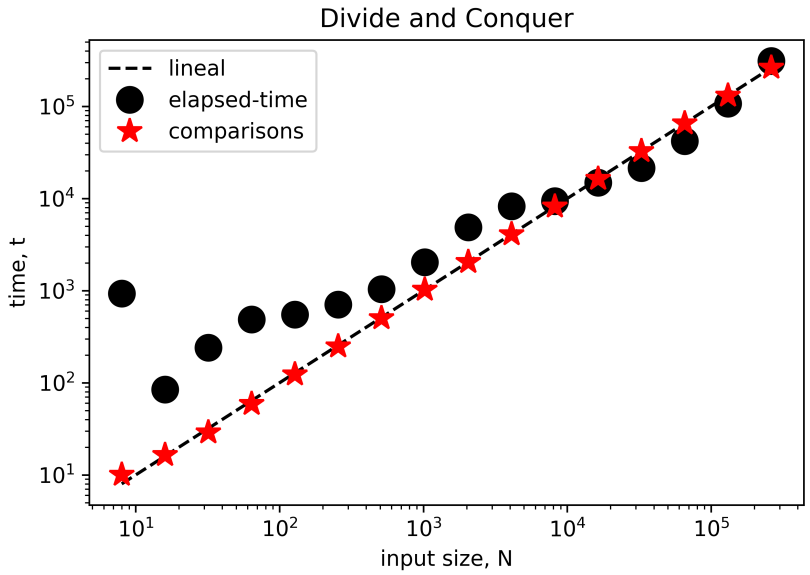

Figure 2: Divide and Conquer Graph

4.2 Program with Lists

Brute Force:

N	Comparisons	Time execution (NanoSeconds)
2	5	80285
4	21	17141
8	78	64173
16	286	136374
32	2717	391198
64	22107	2267096
128	232501	6677162
256	1889827	7221301
512	14383399	39814464
1024	120355224	430530121
2048	949170608	3618172341
4096	7658326502	29357727502
8192	61112878563	277248164117
16384	484881686078	2433185704446

Table 3: Brute Force Results

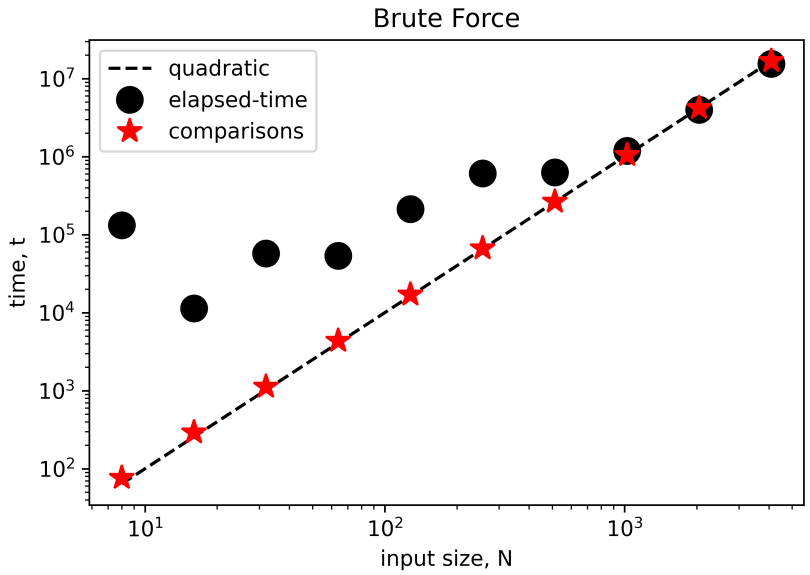

Figure 3: Brute Force Graph

Divide and Conquer:

N	Comparisons	Time execution (NanoSeconds)
2	6	26089
4	25	57323
8	53	71927
16	118	84348
32	1816	458970
64	18255	2205701
128	216610	5971473
256	1825336	6870489
512	14123271	38353348
1024	119310831	423915268
2048	944984676	3589495232
4096	948368697	29245593041
8192	916260924	276782795586
16384	717986884	2431116469010

Table 4: Divide and Conquer Results

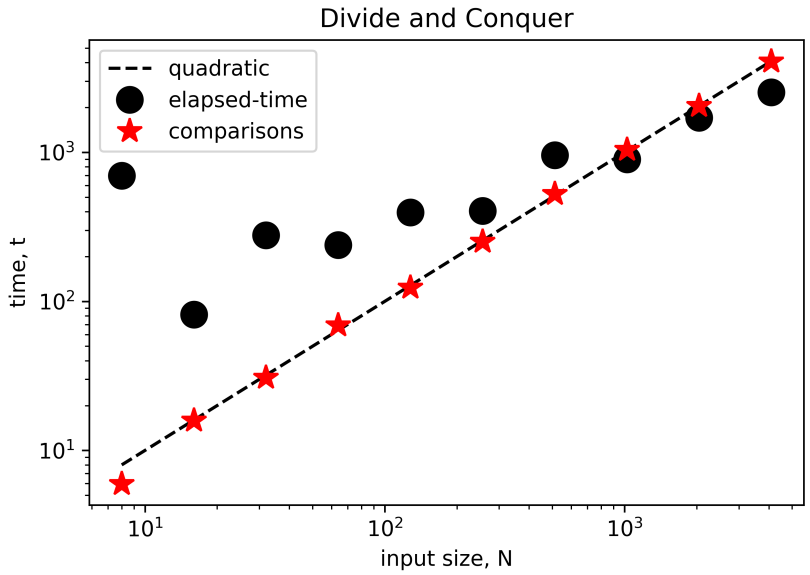

Figure 4: Divide and Conquer Graph

5 Discussion

The results show that the complexity of the brute force and the divide and conquer function stays the same by using linked lists. For the brute force the complexity is quadratic and for the divide and conquer function is linear. But, by using linked list it was required to decrease the n input as you can see in Table 3 and Table 4 because it took alot more time than the program that used ArrayList. Even though the amount of coordinates was decreased for the program with Linked List the time of execution was very similar to the program with ArrayList. This indicates that ArrayList are more efficient because they take less time with more coordinates than linked lists. In addition, we also notice that Brute force works better with less amount of coordinates and the divide and conquer function is more useful for a bigger amount of coordinates. These is evidenced in the tables, as you can see with a smaller number of coordinates the brute force has less number of comparisons that the divide and conquer but when the number of coordinates increases divide and conquer has less comparisons.

6 Conclusions

To conclude, the purpose of this experiment is to explain what data structure is more ideal to solve the closest pair problem. In this case it was proven that using ArrayList works better because ArrayList have a more efficient way of getting sorted and finding an especific

coordinate in the ArrayList. Even though both can be used to solve the problem ArrayLists simplify the solution. To improve the experiment we can test different sorting algorithms for the Linked List program.

References

[1] Misael Diaz Maldonado. *loglogPlot.py*. 2022. URL: //github.com/misael-diaz/computer-programming/%20blob/main/src/io/java/loglogPlot.py%22.