# **TITLE (A SHORT DESCRIPTION OF THE PROJECT, BEWEEN 8 AND 12 WORDS**

| Name of first author  University (name in Spanish)  Country  E-mail at Eafit | Name of second author  University (name in Spanish)  Country  E-mail at Eafit | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |
| --- | --- | --- |

**NOTE: To have more information about the sections in this report, please read “*Guía para la realización del Proyecto Final de Estructura de Datos 1.”* For the final version of this report: 1. Detele all text in red. 2. Adjust spaces among words and paragraphs. 3. Change the color of all the texts to black. You should also keep in mind the meaning of the colors:**

**Black text** = To complete for the 1st deliverable

Blue text = To complete for the 2nd deliverable

Violet text = To complete for the 3rd deliverable

# **ABSTRACT**

To write the abstract, you should answer the following questions in a paragraph: What is the problem? Why is the problem important? Which are the related problems? What is the solution you proposed?, what results did you achieve? , What are the conclusions of this work? Abstract should have at most 200 words.

## **Keywords**

|  |
| --- |
| Keywords that you consider meaningfull to index this report  in libraries and data bases. |

## **ACM CLASSIFICATION Keywords**

|  |
| --- |
| Only keywords in the ACM classification which can be  found at <http://bit.ly/2oVE5 2i>  You cannot create your own keywords here. |

# Example:Theory of computation → Design and analysis of algorithms → Graph algorithms analysis → Shortest paths

# **1. INTRODUCTION**

Is the motivation in the real world that leads to the problem. This means, include some history of this problem.

# **2. PROBLEM**

In a few words, explain the problem, the impact that has in society and why is important to solve the problem.

## **3. RELATED WORK**

## Explain 4 algorithmic problems related to the problem described in Section 2. You may find the related problems in book, scientific articles or websites. You should explain at least one solution for each problem. DO NOT include technological solutions, focus on algorithmic problems and algorithmic solutions.

## **3.1 Title of the first related problem**

You should mention the first algorithmic problem and its solution.

## **3.2 Title of the second related problem**

You should mention the second algorithmic problem and its solution.

## **3.3 Title of the third related problem**

You should mention the third algorithmic problem and its solution.

## **3.4 Title of the fourth related problem**

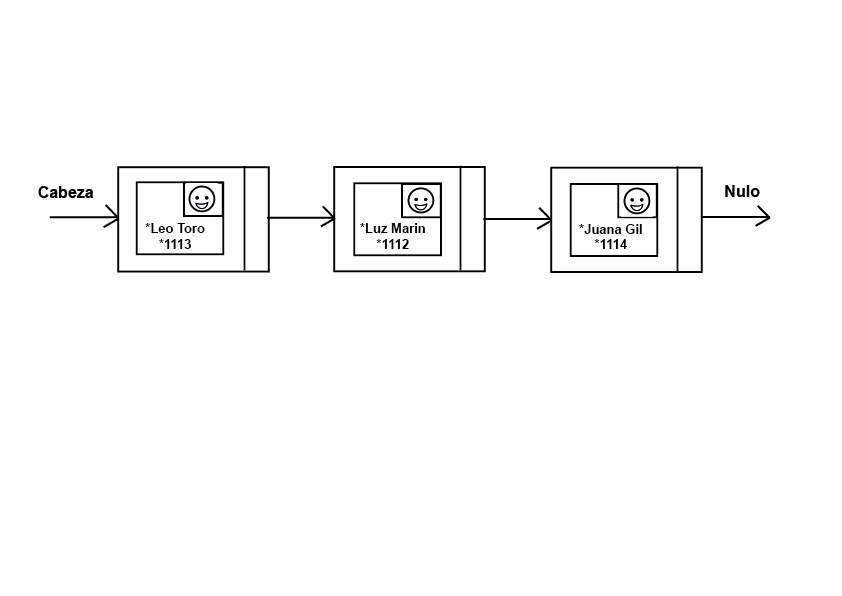
**You should mention the fourth algorithmic problem and its solution.**

## **4. TITLE OF THE FIRST ALGORITHM DESIGNED**

## In what follows we explain the data structure and the algorithm.

## **4.1 Data Structure**

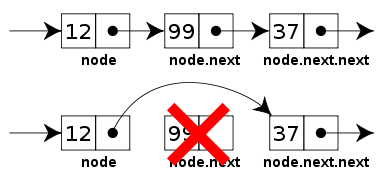
## Design a data structure to solve the problem and make a figure explaining it. Do not use figures from the Internet.



**Figure 1:** Linked List of persons. Una person is a class that contains a name, id number and photo.

## **4.2 Operations of the data structure**

Design the oepration of the data structure to solve the problem efficiently. Include one figure to explain each operation.



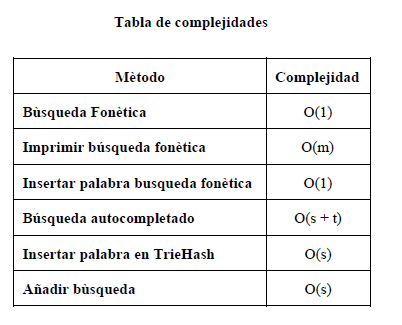
**Figure 2:** Delete operation of a Linked List.

**4.3 Design criteria of the data structure**

Explain objetive criteria that you considered to design the data structure. Examples of objective criteria are efficiency in time and space. Non-objective criteria will lower your grade. Examples of non-objetive criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grade.

**4.4 Complexity analysis**

Derive the complexity of each operation of the data structure for the worst case and best case, As an example, this is a way to report the complexity analysis:



**Table 1:** Table to report complexity analysis

## **4.5 Execution time**

Measure (I) execution time and (II) memory used by the operations of the data structure, for the data set found in the .ZIP file.

## Measure the execution time and memory used 100 for each data set and for each operation of the data structure. Report the average values.

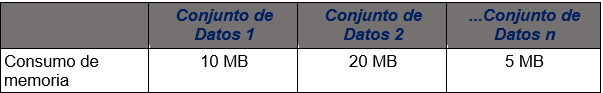
|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Dataset 1*** | ***Dataset 2*** | ***...Dataset n*** |
| *Best case* | 10 s | 20 s | 5 s |
| *Average case* | 12 s | 10 s | 35 s |
| *Worst case* | 15 s | 21 s | 35 s |

## 

## **Table 2:** Execution time of the operations of the data structure for each data set.

## **4.6 Memory consumption**

Report the memory used for each data set

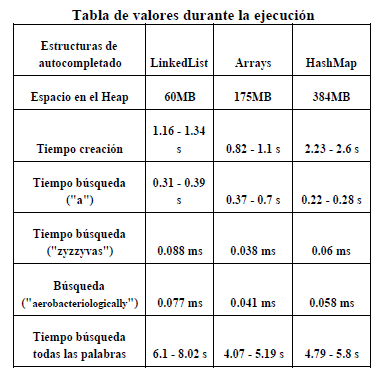


## **Table 3:** Memory used for each operation of the data structure and for each data set data sets.

## To measure memory consumption, you should use a profiler. An very good one for Java is VisualVM, developed by Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>

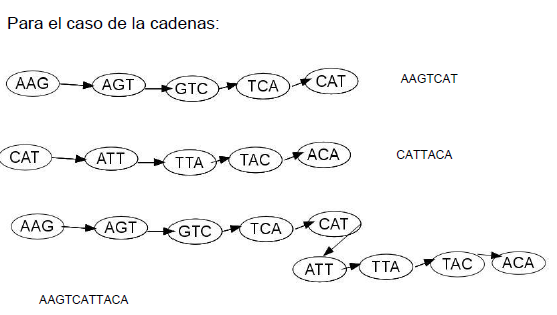
## **4.7 Result analysis**

Explain the results obtained. As an example, compare different implementation of the data structure and report the comparison in a table or graph.



**Table 4:** Analysis of the results

**4.7 Algorithm**

Design one algorithm to solve the problem and make a figure. Do not use figures from the Internet, make your own.

**Figure 3:** Step by step explaining how to assemble DNA fragments using Bruijn graphs.

**4.8** **Complexity analysis of the algorithm**

Calculate the complexity of the algorithm for the worst case, best case and average case.

|  |  |
| --- | --- |
| **Subproblems** | **Complexity** |
| Read the DNA sequences | O(S) |
| Update *de Bruijn* graph with the DNA sequences | O(A.V2) |
| Traverse *de Bruijn* graph to create a new DNA sequence | O(S) |
| **Total complexity** | O(A.V2 + S) |

**Table 2:** Complexity of each subproblem that is part of the algorithm. Let A be the length of a DNA segment, S the number of DNA segments and V the number of different k-mers obtained from the DNA sequences.

NOTE: Without total complexity, this analysis is useless.

**4.7 Design criteria of the algorithm**

Explain why the algorithm was design that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory consumption. Examples of non-objetive criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grade.

**4.8 Execution times**

Compute, execution time for each dataset in the ZIP file

## Measure execution time 100 times for each dataset and report average execution time.

## 

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Dataset 1*** | ***Dataset 2*** | ***...Dataset n*** |
| *Best case* | 10 s | 20 s | 5 s |
| *Average case* | 12 s | 10 s | 35 s |
| *Worst case* | 15 s | 21 s | 35 s |

## **Table 3:** Execution time of the algorithm for different datasets.

## **4.9 Memory consumption**

Measure memory consumption of the algorithm for different datasets

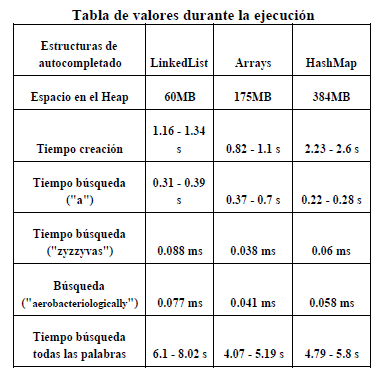
|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Dataset 1*** | ***Dataset 2*** | ***...Dataset n*** |
| **Memory consumption** | 10 MB | 20 MB | 5 MB |

## **Tabla 4:** Memory consumption of the algorithm for different datasets.

## To measure memory consumption, you should use a profiler. An very good one for Java is VisualVM, developed by Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>

## **4.10 Analysis of the results**

Explain the results obtained. Make a table or a graph explaining the time and memory consumption.



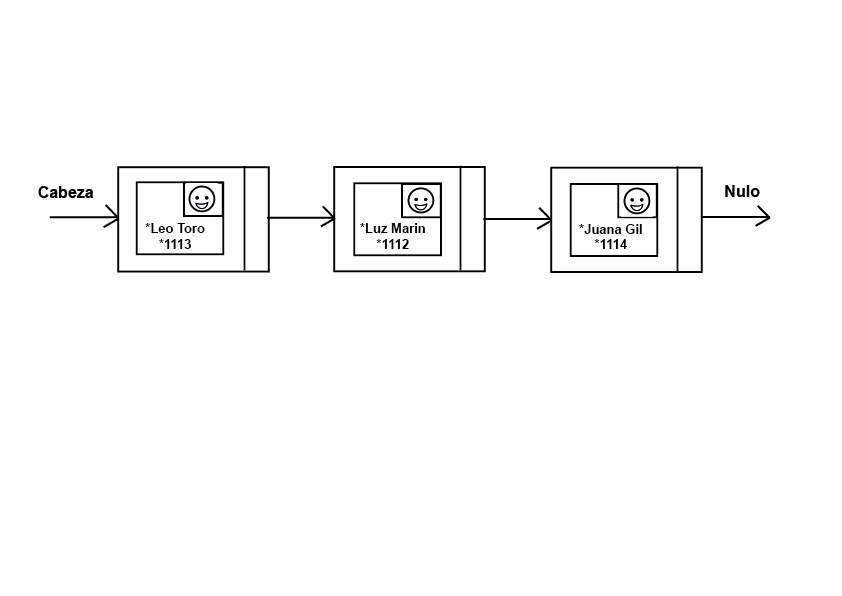
**Table 5:** Analysis of the results obtained from the algorithm execution.

**5. TITLE OF THE FINAL SOLUTION DESIGNED**

## In what follows we explain the data structure and algorithms for this solution.

## **5.1 Data Structure**

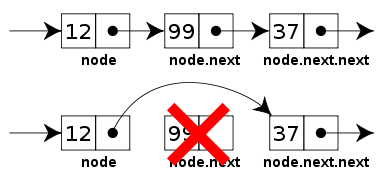
## Design a data structure to solve the problem and make a figure explaining it. Do not use figures from the Internet.



**Figure 1:** Linked List of persons. Una person is a class that contains a name, id number and photo.

## **5.2 Operations of the data structure**

Design the oepration of the data structure to solve the problem efficiently. Include one figure to explain each operation.



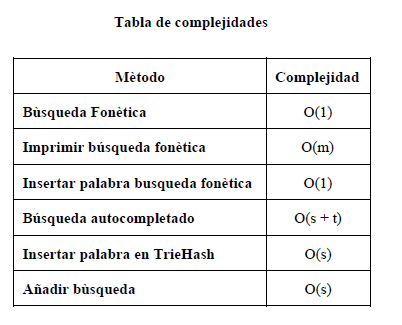
**Figure 2:** Delete operation of a Linked List.

**5.3 Design criteria of the data structure**

Explain objetive criteria that you considered to design the data structure. Examples of objective criteria are efficiency in time and space. Non-objective criteria will lower your grade. Examples of non-objetive criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grade.

**5.4 Complexity analysis**

Derive the complexity of each operation of the data structure for the worst case and best case, As an example, this is a way to report the complexity analysis:



**Table 5:** Table to report complexity analysis

## **5.5 Execution time**

Measure (I) execution time and (II) memory used by the operations of the data structure, for the data set found in the .ZIP file.

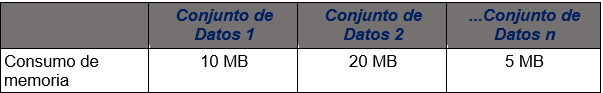
## Measure the execution time and memory used 100 for each data set and for each operation of the data structure. Report the average values.

## 

## **Table 6:** Execution time of the operations of the data structure for each data set.

## **5.6 Memory used**

Report the memory used for each data set

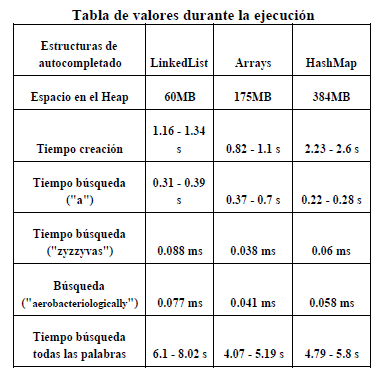


## **Table 7:** Memory used for each operation of the data structure and for each data set data sets.

## To measure memory consumption, you should use a profiler. An very good one for Java is VisualVM, developed by Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>

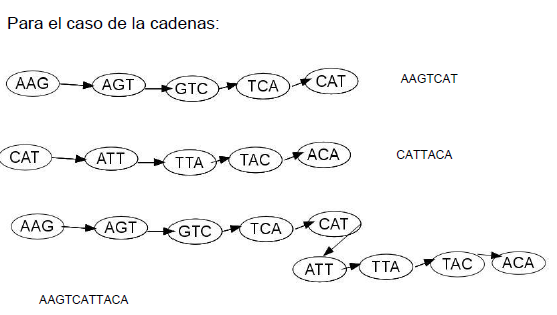
## **5.7 Result analysis**

Explain the results obtained. As an example, compare different implementation of the data structure and report the comparison in a table or graph.



**Table 8:** Analysis of the results

**5.8 Algorithm**

Design one algorithm to solve the problem and make a figure. Do not use figures from the Internet, make your own.

**Figure 6:** Step by step explaining how to assemble DNA fragments using *de Bruijn* graphs.

**5.9** **Complexity analysis of the algorithm**

Calculate the complexity of the algorithm for the worst case, best case and average case.

|  |  |
| --- | --- |
| **Subproblems** | **Complexity** |
| Read the DNA sequences | O(S) |
| Update *de Bruijn* graph with the DNA sequences | O(A.V2) |
| Traverse *de Bruijn* graph to create a new DNA sequence | O(S) |
| **Total complexity** | O(A.V2 + S) |

**Table 8.** Complexity of each subproblem that is part of the algorithm. Let A be the length of a DNA segment, S the number of DNA segments and V the number of different k-mers obtained from the DNA sequences.

NOTE: Without total complexity, this analysis is useless.

**5.10 Design criteria of the algorithm**

Explain why the algorithm was design that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory consumption. Examples of non-objetive criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grade.

.

**5.11 Execution times**

Measure (I) execution time and (II) memory used by the operations of the data structure, for the data set found in the .ZIP file.

## Measure the execution time and memory used 100 for each data set and for each operation of the data structure. Report the average values.

## 

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Dataset 1*** | ***Dataset 2*** | ***...Dataset n*** |
| *Best case* | 10 s | 20 s | 5 s |
| *Average case* | 12 s | 10 s | 35 s |
| *Worst case* | 15 s | 21 s | 35 s |

## **Tabla 9.** Execution time of the operations of the data structure for each data set.

## To measure memory consumption, you should use a profiler. An very good one for Java is VisualVM, developed by Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>

## **5.12 Memory consumption**

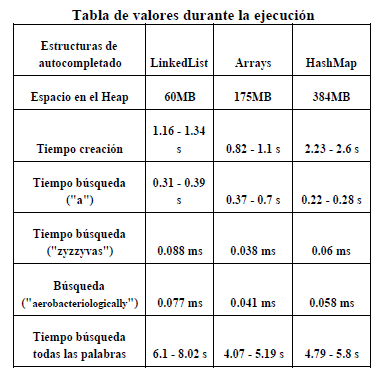
Report the memory used for each data set

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Dataset 1*** | ***Dataset 2*** | ***...Dataset n*** |
| **Memory consumption** | 10 MB | 20 MB | 5 MB |

## **Tabla 10.** Consumo de memoria del algoritmo con diferentes conjuntos de datos

## **5.13 Analysis of the results**

Explain the results obtained. As an example, compare different implementation of the data structure and report the comparison in a table or graph.



**Table 11.** Analysis of the results

**6. CONCLUSIONS**

To write the conclusions, proceed in the following way. 1. Write a paragraph with a summary, the most important issued of the report. 2. In another paragraph, explain the most important results that you obtained with the last data structure you designed. 3. Compare your first solution with the last solution. 4. At last, explain future work, a future continuation of this project. You can also mention in the conclusions, technical problems that you had during the development of the data structure and its implementation and you solved them.

**6.1 Future work**

Answer, what would you like to improve in the future? How would you like to improve your data structure and its implementation?

# **ACKNOWLEDGEMENTS**

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention websites of authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

internet ni autores de artículo leídos con quienes no se han contactado. 3. Los nombres que sí van son quienes ayudaron, compañeros del curso o docentes de otros cursos.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

# **REFERENCES**

Reference sourced using ACM reference format. Read ACM guidelines in <http://bit.ly/2pZnE5g>

As an example, consider this two references:

1.Adobe Acrobat Reader 7, Be sure that the references sections text is Ragged Right, Not Justified. <http://www.adobe.com/products/acrobat/>.

2. Fischer, G. and Nakakoji, K. Amplifying designers’ creativity with domainoriented design environments. in Dartnall, T. ed. Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, Dordrecht, 1994, 343-364.