# **LOS ALGORITMOS COMO SOLUCIÓN AL TRAFICO**

| David Jose Cardona Nieves  Universidad Eafit  Colombia  djcardonan@eafit.edu.co | Juan David Valencia Torres  Universidad Eafit  Colombia  jdvalencit@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |
| --- | --- | --- |

**Textos en negro** = Es todo lo que deben hacer en la entrega 1

**Textos en Verde** = Es todo lo que deben hacer en la Entrega 2

Textos en violeta = Es todo lo que deben hacer en la entrega 3

# **RESUMEN**

El enigma a resolver dentro de la realización de este proyecto es el poder encontrar cómo reducir la emisión de gases de efecto invernadero, un tema ya bastante preocupante a día de hoy. Para dar solución a este problema se plantea el uso de algoritmos en la realización de rutas eficientes tanto a nivel productivo como ecológico. Sin embargo, este proyecto se encuentra con un gran limitante (y principalmente, un gran reto) y este es el no poder establecer las rutas de forma casi apoteósica debido a la restricción de movilidad que presentan los vehículos por su necesidad de recargar batería. El resultado a esperar es el poder tener un algoritmo eficiente que pueda compararse con otros realizados en ejercicios alternos como eVRP.

## **Palabras clave**

|  |
| --- |
| Estas son palabras claves que ustedes consideran apropiadas  para indexar el informe PDF en bibliotecas o bases de datos |

## **Palabras clave de la clasificación de la ACM**

|  |
| --- |
| Solo las que están en esta lista de ACM, en [http://bit.ly/2oVE5 2i](http://bit.ly/2oVE5%202i)  No pueden ser inventadas por ustedes. |

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

# **1. INTRODUCCIÓN**

No podemos negar que el ámbito ecológico ha ido ganando poder en nuestro querer, hacer y pensar con el pasar del tiempo. Es así como han venido forjándose en la mente de millones de personas ideas en pro de un mejor cuidado de el medio ambiente.

Cada día podemos ver como nacen y crecen nuevas tecnologías como los vehículos eléctricos. Sin embargo, estas tecnologías requieren ser reguladas y mejoradas con el pasar de el tiempo. Por poner un ejemplo, podemos hablar de los vehículos eléctricos y el cómo podríamos aportar mucho al medio ambiente si hacemos de esta una tecnología eficiente.

# **2. PROBLEMA**

Hemos de anticiparnos al futuro buscando resolver la gran incógnita de este siglo: ¿Cómo mantener el equilibrio entre vida y desarrollo?

Más entrados en detalles, podremos hablar de cómo establecer de forma precisa y eficiente las rutas a ser usadas por parte de vehículos de carga, transporte, etc… Para llegar a un destino recorriendo cada uno de los lugares que le digamos que recorra. Llevando esto a la realidad, hablaríamos de cómo un hipotético vehículo autónomo podría recoger personal de una empresa en sus respectivas direcciones para llevarlos al trabajo, por poner un ejemplo.

**3. TRABAJOS RELACIONADOS** Aquí deberán explicar 4 problemas algorítmicos similares que se encuentren documentados en libros, artículos científicos o sitios web, y dar al menos 1 solución para uno de ellos. NO poner soluciones de tecnología.

3.1 **Resolución del problema de enrutamiento de vehículos con limitaciones de capacidad utilizando un procedimiento metaheurístico de dos fases** Este documento plantea una solución al problema de enrutamiento de vehículos aplicando lo que ellos conocen como “límites de flujo”, dividiendo este en dos fases: la primera donde se realiza el proceso de diseño de rutas y la segunda fase donde se realiza la planificación de la flota.

3.2 **A Vehicle Routing Problem with a Time Windows Approach to Improve the Delivery Process** Pineda y Carabalí buscan resolver el VRP (Ejercicio propuesto). Esto lo hacen basándose en el algoritmo de la Heurística Estocástica de Clarke-Wright. Este algoritmo ha de tener en su mayoría de casos de implementación, una complejidad de O(n^3)

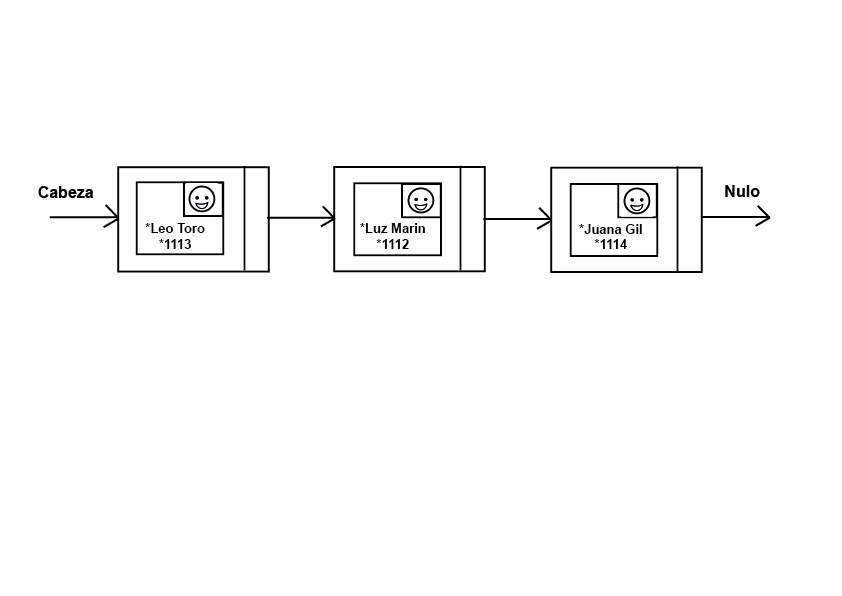
3.3 **Modelo de ruteo de vehículos para la distribución de las empresas Laboratorios Veterland, Laboratorios Callbest y Cosméticos Marlioü París** La temática abordada por Andrés Felipe en su artículo, consta de plantear un modelo de ruteo para distribución de productos, planteando este mismo en dos fases: una inicial en la que se esboza el mapa y una segunda para generar todas las rutas. El algoritmo utilizado para darle solución a este problema fue el de Tabu, que platea una posible solución y posteriormente busca una mejor, si es el caso. Los resultados obtenidos priorizan un menor tiempo en cada recorrido

## **4. TÍTULO DE LA PRIMERA SOLUCIÖN DISEÑADA**

## A continuación, explicamos la estructura de datos y el algoritmo.

## **4.1 Estructura de datos**

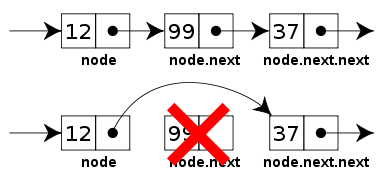
## Diseñen la estructura de datos para resolver el problema y grafíquenla. No usar gráficas extraídas de internet



## **Gráfica 1:** Lista simplemente encadenada de personas. Una persona es una clase que contiene nombre, cédula y foto

## **4.2 Operaciones de la estructura de datos**

## Diseñen las operaciones de la estructura de datos para solucionar el problema eficientemente. Incluyan una imagen explicando cada operación



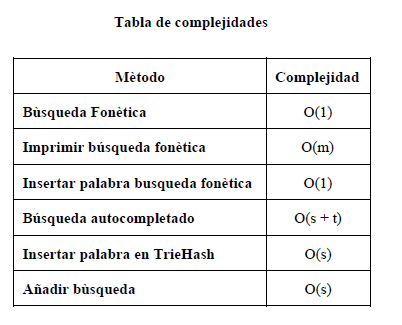
**Gráfica 2:** Imagen de una operación de borrado de una lista encadenada

## **4.3 Criterios de diseño de la estructura de datos**

Expliquen con criterios objetivos, por qué diseñaron así la estructura de datos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaulación con 40%

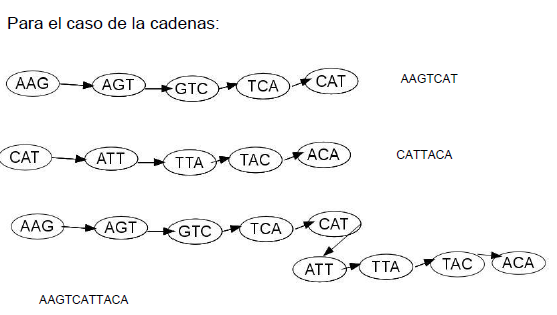
**4.4 Análisis de Complejidad**

Calculen la complejidad de las operaciones de la estructura de datos para el peor de los casos. Vean un ejemplo para reportarla:



**Tabla 1:** Tabla para reportar la complejidad

**4.5 Algoritmo**

Diseñen el algoritmo para resolver el problema y grafíquenlo. No usen gráficas extraídas de internet

**Gráfica 3:** Paso a paso cómo se ensamblan fragmentos de ADN utilizando los grafos de *Bruijn*.

**4.6** **Cálculo de la complejidad del algoritmo**

Calculen la complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio

|  |  |
| --- | --- |
| **Sub problema** | **Complejidad** |
| Crear el grafo de *Bruijn* con las secuencias | O(S) |
| Actualizar el grafo de *Bruijn* con las secuencias | O(A.V2) |
| Encontrar los genes | O(S) |
| **Complejidad Total** | O(A.S2 + V) |

**Tabla 2:** complejidad de cada uno de los sub problemas que componen el algoritmo. Sea A la longitud de una secuencia de ADN, S el número de secuencias de ADN, y V el número de K-meros diferentes que se obtienen de las secuencias de ADN.

NOTA: Sin complejidad total, el análisis no sirve de nada

**4.7 Criterios de diseño del algoritmo**

Expliquen por qué diseñaron así el algoritmo. Usen criterios objetivos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaluación con 40%.

**4.8 Tiempos de Ejecución**

Calculen, (I) el tiempo de ejecución y (II) la memoria usada del algoritmo, para el Conjunto de Datos que está en el ZIP:

## Tomen 100 veces el tiempo de ejecución y memoria de ejecución, para cada conjunto de datos

## 

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Conjunto de Datos 1*** | ***Conjunto de Datos 2*** | ***...Conjunto de Datos n*** |
| *Mejor caso* | 10 sg | 20 sg | 5 sg |
| *Caso promedio* | 12 sg | 10 sg | 35 sg |
| *Peor caso* | 15 sg | 21 sg | 35 sg |

## **Tabla 3:** Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

## **4.9 Memoria**

Mencionar la memoria que consume el programa para varios ejemplos

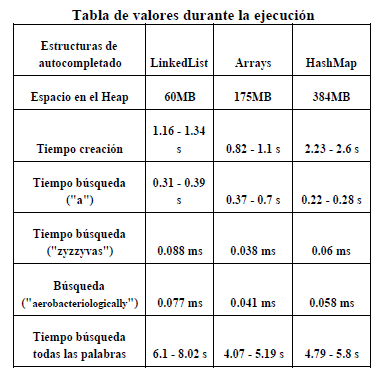
|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Conjunto de Datos 1*** | ***Conjunto de Datos 2*** | ***...Conjunto de Datos n*** |
| **Consumo de memoria** | 10 MB | 20 MB | 5 MB |

## **Tabla 4:** Consumo de memoria del algoritmo con diferentes conjuntos de datos

## Para medir la memoria que consume un programa, se utilizan generadores de perfiles (en Inglés, profilers). Uno muy bueno para Java es VisualVM, desarrollado por Oracle,<http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html> No dejen de usarlo en sus proyectos y en la vida. Para usarlo hay que generar un .jar que es como un ejecutable de Java. En Netbeans "martillo con escoba" y en BlueJ "archivo, generar .jar".

## **4.10 Análisis de los resultados**

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:



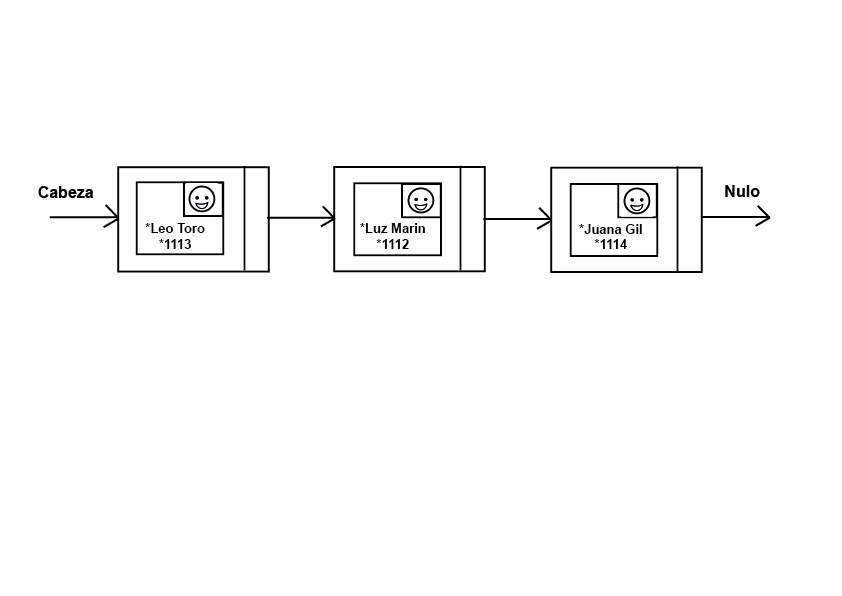
**Tabla 5:** Análisis de los resultados obtenidos con la implementación del algoritmo

**5. TÍTULO DE LA SOLUCIÓN FINAL DISEÑADA**

## A continuación, explicamos la estructura de datos y el algoritmo.

## **5.1 Estructura de datos**

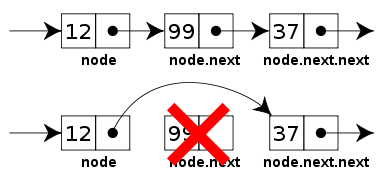
## Diseñen la estructura de datos para resolver el problema y grafíquenla. No usar gráficas extraídas de internet



## **Gráfica 4:** Lista simplemente encadenada de personas. Una persona es una clase que contiene nombre, cédula y foto

## **5.2 Operaciones de la estructura de datos**

## Diseñen las operaciones de la estructura de datos para solucionar el problema eficientemente. Incluyan una imagen explicando cada operación



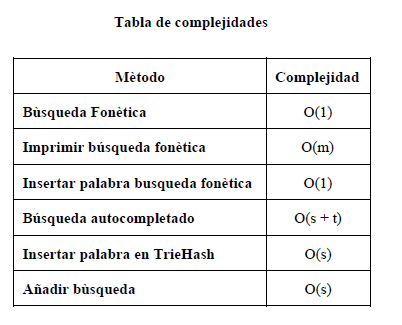
**Gráfica 5:** Imagen de una operación de borrado de una lista encadenada

## **5.3 Criterios de diseño de la estructura de datos**

Expliquen con criterios objetivos, por qué diseñaron así la estructura de datos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaluación con 40%

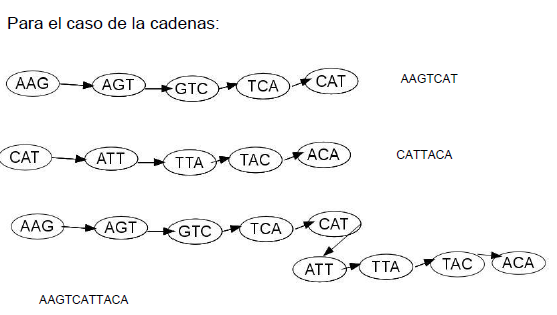
**5.4 Análisis de Complejidad**

Calculen la complejidad de las operaciones de la estructura de datos para el peor de los casos. Vean un ejemplo para reportarla:



**Tabla 6:** Tabla para reportar la complejidad

**5.5 Algoritmo**

Diseñen el algoritmo para resolver el problema y grafíquenlo. No usen gráficas extraídas de internet

**Gráfica 6:** Paso a paso cómo se ensamblan fragmentos de ADN utilizando los grafos de *Bruijn*.

**5.6** **Cálculo de la complejidad del algoritmo**

Calculen la complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio

|  |  |
| --- | --- |
| **Sub problema** | **Complejidad** |
| Crear el grafo de *Bruijn* con las secuencias | O(N) |
| Actualizar el grafo de *Bruijn* con las secuencias | O(A.N2) |
| Encontrar los genes | O(V) |
| **Complejidad Total** | O(A.N2 + V) |

**Tabla 7:** complejidad de cada uno de los sub problemas que componen el algoritmo. Sea A la longitud de una secuencia de ADN, N el número de secuencias de ADN, y V el número de K-meros diferentes que se obtienen de las secuencias de ADN.

**5.7 Criterios de diseño del algoritmo**

Expliquen por qué diseñaron así el algoritmo. Usen criterios objetivos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaluación con 40%

.

**5.8 Tiempos de Ejecución**

Calculen, (I) el tiempo de ejecución y (II) la memoria usada del algoritmo, para el Conjunto de Datos que está en el ZIP:

## Tomen 100 veces el tiempo de ejecución y memoria de ejecución, para cada conjunto de datos

## 

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Conjunto de Datos 1*** | ***Conjunto de Datos 2*** | ***...Conjunto de Datos n*** |
| *Mejor caso* | 10 sg | 20 sg | 5 sg |
| *Caso promedio* | 12 sg | 10 sg | 35 sg |
| *Peor caso* | 15 sg | 21 sg | 35 sg |

## **Tabla 8:** Tiempos de ejecución del algoritmo con diferentes conjuntos de datos

## Para medir la memoria que consume un programa, se utilizan generadores de perfiles (en Inglés, profilers). Uno muy bueno para Java es VisualVM, desarrollado por Oracle,<http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html> No dejen de usarlo en sus proyectos y en la vida. Para usarlo hay que generar un .jar que es como un ejecutable de Java. En Netbeans "martillo con escoba" y en BlueJ "archivo, generar .jar".

## **5.9 Memoria**

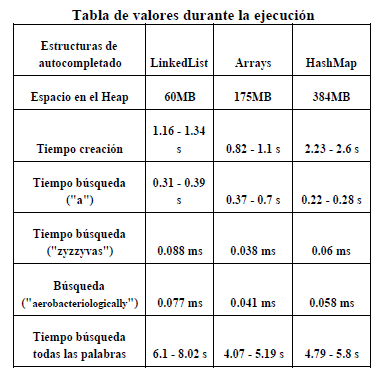
Mencionar la memoria que consume el programa para varios ejemplos

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Conjunto de Datos 1*** | ***Conjunto de Datos 2*** | ***...Conjunto de Datos n*** |
| **Consumo de memoria** | 10 MB | 20 MB | 5 MB |

## **Tabla 9:** Consumo de memoria del algoritmo con diferentes conjuntos de datos

## **5.10 Análisis de los resultados**

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:



**Tabla 10:** Análisis de los resultados obtenidos con la implementación del algoritmo

**6. CONCLUSIONES**

Para escribirlas, procedan de la siguiente forma: 1. En un párrafo escriban un resumen de lo más importante que hablaron en el reporte. 2. En otro expliquen los resultados más importantes, por ejemplo, los que se obtuvieron con la solución final. 3.Luego, comparen la primera solución que hicieron con los trabajos relacionados y la solución final. 4. Por último, expliquen los trabajos futuros para una posible continuación de este Proyecto. Aquí también pueden mencionar los problemas que tuvieron durante el desarrollo del proyecto

**6.1 Trabajos futuros**

Respondan ¿Qué les gustaría mejorar en el futuro? ¿Qué les gustaría mejor al algoritmo, estructura de datos, implementación?

# **AGRADECIMIENTOS**

Identifiquen el tipo de agradecimiento que van a escribir: para una persona o para una institución. Luego, escríbanlo de acuerdo al idioma y tengan en cuenta que: 1. El nombre del docente no va porque él es autor. 2. Tampoco sitios de 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.

Aquí un ejemplo en inglés: 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.

# **REFERENCIAS**

Referenciar las fuentes usando el formato para referencias de la ACM. Léase en <http://bit.ly/2pZnE5g> Vean un ejemplo:

1. Velandia, J. Felipe. 2018. Propuesta de un modelo de ruteo VRP para una empresa OPL de medicamentos y suministros médicos en Bogotá.

2. Daza, J. Montoya, J. and Narducci, 2009. RESOLUCIÓN DEL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS CON LIMITACIONES DE CAPACIDAD UTILIZANDO UN PROCEDIMIENTO METAHEURÍSTICO DE DOS FASES. 12th ed. Medellín: Escuela de Ingeniería de Antioquia, pp. 23-34.

3. Carabalí, H. and Pineda, U. 2020. Un Problema de Enrutamiento del Vehículo con Enfoque de Ventanas de Tiempo para Mejorar el Proceso de Entregas. Ingeniería