REPORTE TAREA 2
ALGORITMOS Y COMPLEJIDAD

«Explorando la Distancia entre Cadenas, una Operación a la Vez»

Al Goritmo Pérez

2 de octubre de 2024

23:39

Resumen

Un resumen es un breve compendio que sintetiza todas las secciones clave de un trabajo de investigación: la introducción, los objetivos, la infraestructura y métodos, los resultados y la conclusión. Su objetivo es ofrecer una visión general del estudio, destacando la novedad o relevancia del mismo, y en algunos casos, plantear preguntas para futuras investigaciones. El resumen debe cubrir todos los aspectos importantes del estudio para que el lector pueda decidir rápidamente si el artículo es de su interés.

En términos simples, el resumen es como el menú de un restaurante que ofrece una descripción general de todos los platos disponibles. Al leerlo, el lector puede hacerse una idea de lo que el trabajo de investigación tiene para ofrecer [1].

La extensión del resumen, para esta entrega, debe ser tal que la totalidad del índice siga apareciendo en la primera página. Recuerde que NO puede modificar el tamaño de letra, interlineado, márgenes, etc.

Índice

1. Introducción

La extensión máxima para esta sección es de 2 páginas.

La introducción de este tipo de informes o reportes, tiene como objetivo principal **contextualizar el problema que se va a analizar**, proporcionando al lector la información necesaria para entender la relevancia del mismo.

Es fundamental que en esta sección se presenten los antecedentes del problema, destacando investigaciones previas o principios teóricos que sirvan como base para los análisis posteriores. Además, deben explicarse los objetivos del informe, que pueden incluir la evaluación de un algoritmo, la comparación de métodos o la validación de resultados experimentales.

Aunque la estructura y el enfoque siguen principios de trabajos académicos, se debe recordar que estos informes no son publicaciones científicas formales, sino trabajos de pregrado. Por lo tanto, se busca un enfoque claro y directo, que permita al lector comprender la naturaleza del problema y los objetivos del análisis, sin entrar en detalles excesivos.

Introduction Checklist de *How to Write a Good Scientific Paper* [7], adaptada a nuestro contexto:

- Indique el **campo del trabajo** (Análisis y Diseño de algoritmos en Ciencias de la Computación), por qué este campo es importante y qué se ha hecho ya en este área, con las **citas** adecuadas de la literatura académica o fuentes relevantes.
- Identifique una **brecha** en el conocimiento, un desafío práctico, o plantee una **pregunta** relacionada con la eficiencia, complejidad o aplicabilidad de un algoritmo particular.
- Resuma el propósito del informe e introduzca el análisis o experimento, dejando claro qué se está investigando o comparando, e indique **qué es novedoso** o por qué es significativo en el contexto de un curso de pregrado.
- Evite; repetir el resumen; proporcionar información innecesaria o fuera del alcance de la materia (limítese al análisis de algoritmos o conceptos de complejidad); exagerar la importancia del trabajo (recuerde que se trata de un informe de pregrado); afirmar novedad sin una comparación adecuada con lo enseñado en clase o la bibliografía recomendada.

Recuerde que este es su trabajo, y sólo usted puede expresar con precisión lo que ha aprendido y quiere transmitir. Si lo hace bien, su introducción será más significativa y valiosa que cualquier texto automatizado. ¡Confíe en sus habilidades, y verá que puede hacer un mejor trabajo que cualquier herramienta que automatiza la generación de texto!

2. Diseño y Análisis de Algoritmos

La extensión máxima para esta sección es de 4 páginas.

Diseñar un algoritmo por cada técnica de diseño de algoritmos mencionada en la sección de objetivos. Cada algoritmo debe resolver el problema de distancia mínima de edición extendida, dadas dos cadenas `S1` y `S2`, utilizando las operaciones y costos especificados.

- Describir la solución diseñada.
- Incluir pseudocódigo (ver ejemplo algoritmo 1)
- Proporciones un ejemplo paso a paso de la ejecución de sus algoritmos que ilustren cómo sus algoritmos manejan diferentes escenarios, particularmente donde las transposiciones o los costos variables afectan el resultado. Haga referencias a los programas expresados en psudocódigo (además puede hacer diagramas).
- Analizar la Complejidad temporal y espacial de los algoritmos diseñados en términos de las longitudes de las cadenas de entrada $S1$ y $S2$
- Discute cómo la inclusión de transposiciones y costos variables impacta la complejidad.

Los pseudocódigos lo he diseñado utilizando el paquete *Algorithm2e documentation* [3] para la presentación de algoritmos. Se recomienda consultar *Algorithm2e on CTAN* [4] y *Writing Algorithms in LaTeX* [8].

Todo lo correspondiente a esta sección es, digamos, en "lapiz y papel", en el sentido de que no necesita de implementaciones ni resultados experimentales.

Recuerde que lo importante es diseñar algoritmos que cumplan con los paradigmas especificados.

Si se utiliza algún código, idea, o contenido extraído de otra fuente, este **debe** ser citado en el lugar exacto donde se utilice, en lugar de mencionarlo al final del informe.

2.1. Fuerza Bruta

> *"Indeed, brute force is a perfectly good technique in many cases; the real question is, can we use brute force in such a way that we avoid the worst-case behavior?"*
>
> — Knuth, 1998 [6]

Algoritmo 1: Este es solo un ejemplo de cómo estructurar el pseudocódigo, con retornos explícitos y llamados a funciones.

```
1  Procedure AlgorithmName(S1, S2)
2      if S1 está vacía then
3          return longitud de S2
4      else if S2 está vacía then
5          return longitud de S1
6      else if S1[0] = S2[0] then
7          return AlgorithmName(S1[1:], S2[1:])
8      else
9          costo ← AuxiliaryFunction(S1, S2)
10         return costo
11 Procedure AuxiliaryFunction(S1, S2)
12     if S1 y S2 son similares then
13         return algún valor o costo
14     else
15         return AlgorithmName(S1 modificado, S2)
```

2.2. Programación Dinámica

> *Dynamic programming is not about filling in tables. It's about smart recursion!*
>
> Erickson, 2019 [2]

1) Describa la solución recursiva.
2) Escriba la relación de recurrencia, incluyendo condiciones y casos base.
3) Identifique subproblemas.
4) Defina estructura de datos a utilizar y especifique el orden de calculo que realiza su programa que utiliza programación dinámica.

Algoritmo 2: Este es solo un ejemplo de cómo estructurar el pseudocódigo, con retornos explícitos y llamados a funciones.

```
1  Procedure AlgorithmName(S1, S2)
2      if S1 está vacía then
3          return longitud de S2
4      else if S2 está vacía then
5          return longitud de S1
6      else if S1[0] = S2[0] then
7          return AlgorithmName(S1[1:], S2[1:])
8      else
9          costo ← AuxiliaryFunction(S1, S2)
10         return costo
11 Procedure AuxiliaryFunction(S1, S2)
12     if S1 y S2 son similares then
13         return algún valor o costo
14     else
15         return AlgorithmName(S1 modificado, S2)
```

3. Implementaciones

4. Experimentos

> "*Non-reproducible single occurrences are of no significance to science.*"
>
> —Popper,2005 [9]

En la sección de Experimentos, es fundamental detallar la infraestructura utilizada para asegurar la reproducibilidad de los resultados, un principio clave en cualquier experimento científico. Esto implica especificar tanto el hardware (por ejemplo, procesador Intel Core i7-9700K, 3.6 GHz, 16 GB RAM DDR4, almacenamiento SSD NVMe) como el entorno software (sistema operativo Ubuntu 20.04 LTS, compilador g++ 9.3.0, y cualquier librería relevante). Además, se debe incluir una descripción clara de las condiciones de entrada, los parámetros utilizados y los resultados obtenidos, tales como tiempos de ejecución y consumo de memoria, que permitan a otros replicar los experimentos en entornos similares. *La replicabilidad es un aspecto crítico para validar los resultados en la investigación científica computacional* [5].

4.1. Dataset (casos de prueba)

4.2. Resultados

En esta sección, los resultados obtenidos, como las gráficas o tablas, deben estar respaldados por los datos generados durante la ejecución de sus programas. Es fundamental que, junto con el informe, se adjunten los archivos que contienen dichos datos para permitir su verificación. Además, se debe permitir y especficiar como obtener esos archivos desde una ejecución en otro computador (otra infraestructura para hacer lso experimentos).

No es necesario automatizar la generación de las gráficas, pero sí es imprescindible que se pueda confirmar que las visualizaciones presentadas son producto de los datos generados por sus algoritmos, aunque la trazabilidad de los datos hasta las visualizaciones es esencial para garantizar que su validez: describa cómo se generaron los datos, cómo se procesaron y cómo se visualizaron de manera que pueda ser replicado por quien lea su informe.

5. Conclusiones

6. Condiciones de entrega

- La tarea se realizará individualmente (esto es grupos de una persona), sin excepciones.
- La entrega debe realizarse vía `http://aula.usm.cl` en un tarball en el área designada al efecto, en el formato `tarea-2-rol.tar.gz` (`rol` con dígito verificador y sin guión).

  Dicho tarball debe contener las fuentes en LaTeX 2ε (al menos `tarea-2.tex`) de la parte escrita de su entrega, además de un archivo `tarea-2.pdf`, correspondiente a la compilación de esas fuentes.
- Si se utiliza algún código, idea, o contenido extraído de otra fuente, este **debe** ser citado en el lugar exacto donde se utilice, en lugar de mencionarlo al final del informe.
- Asegúrese que todas sus entregas tengan sus datos completos: número de la tarea, ramo, semestre, nombre y rol. Puede incluirlas como comentarios en sus fuentes LaTeX (en TeX comentarios son desde % hasta el final de la línea) o en posibles programas. Anótese como autor de los textos.
- Si usa material adicional al discutido en clases, detállelo. Agregue información suficiente para ubicar ese material (en caso de no tratarse de discusiones con compañeros de curso u otras personas).
- No modifique `preamble.tex`, `tarea_main.tex`, `condiciones.tex`, estructura de directorios, nombres de archivos, configuración del documento, etc. Sólo agregue texto, imágenes, tablas, código, etc. En el códigos funte de su informe, no agregue paquetes, ni archivos .tex (a excepción de que agregue archivos en el `tikz`, donde puede agregar archivos .tex con las fuentes de gráficos en `tikz`).
- La entrega debe realizarse dentro del plazo indicado en `http://aula.usm.cl`:

NO SE ACEPTARÁN TAREAS FUERA DE PLAZO.

- Nos reservamos el derecho de llamar a interrogación sobre algunas de las tareas entregadas. En tal caso, la nota de la tarea será la obtenida en la interrogación.

NO PRESENTARSE A UN LLAMADO A INTERROGACIÓN SIN JUSTIFICACIÓN PREVIA SIGNIFICA AUTOMÁTICAMENTE NOTA 0.

A. Apédice 1

Referencias

[1] Elsevier. *Differentiating between an introduction and abstract in a research paper*. Accessed: 2024-10-02. 2024. URL: https://scientific-publishing.webshop.elsevier.com/manuscript-preparation/differentiating-between-and-introduction-research-paper/.

[2] Jeff Erickson. *Algorithms*. Jun. de 2019. ISBN: 978-1-792-64483-2.

[3] Christophe Fiorio. *Algorithm2e documentation*. http://ctan.math.illinois.edu/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf. 2023.

[4] Christophe Fiorio. *Algorithm2e on CTAN*. https://ctan.org/pkg/algorithm2e. 2023.

[5] Jorge Fonseca y Kazem Taghva. «The State of Reproducible Research in Computer Science». En: ene. de 2020, págs. 519-524. ISBN: 978-3-030-43019-1. DOI: 10.1007/978-3-030-43020-7_68.

[6] Donald E. Knuth. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. USA: Addison Wesley Longman Publishing Co., Inc., 1998. ISBN: 0201896850.

[7] Chris Mack y Lola Muxamedova. *How to Write a Good Scientific Paper*. Nov. de 2019.

[8] Overleaf. *Writing Algorithms in LaTeX*. https://www.overleaf.com/learn/latex/Algorithms. 2023.

[9] K. Popper. *The Logic of Scientific Discovery*. Routledge Classics. Taylor & Francis, 2005. ISBN: 9781134470020. URL: https://books.google.cl/books?id=LWSBAgAAQBAJ.