



Mini Proyecto Análisis y Diseño de Algoritmos I

**Profesor: Jesús Alexander Aranda Bueno
Monitor: Samuel Galindo Cuevas**

**12 de noviembre de 2024 (actualización 29
noviembre)**

El proyecto debe realizarse en grupos de máximo 4 personas.

El problema del procesamiento de una encuesta

Una consultoría de datos desea realizar un análisis de los resultados de una gran encuesta, con la finalidad de identificar cuáles temas, con base en ciertas preguntas, tienen opiniones más favorables y poder también responder otras preguntas afines. Esta gran encuesta está dividida en diversos temas, cada tema tiene diferentes preguntas, donde cada pregunta consulta sobre hasta qué punto se está de acuerdo sobre una propuesta; a su vez cada pregunta fue realizada a ciertas personas (encuestados), cada persona solo puede participar respondiendo sólo una pregunta.

La encuesta está organizada en K temas, cada uno de los temas tiene un nombre, además, cada tema consiste de M preguntas, donde cada pregunta puede contar con una cantidad mínima N_{min} y máxima N_{max} de encuestados, cada encuestado tendrá un número como identificador, su nombre, su nivel de experticia sobre la pregunta, cuyo valor es entero entre 0 y 10, y su opinión-respuesta a la pregunta, este último tomará valores enteros de 0 a 10, donde 0 indica que se tiene una opinión completamente desfavorable y 10 una opinión completamente favorable.

K, M, N_{min} y N_{max} son enteros positivos, además $N_{min} < N_{max}$.

La consultoría busca que cada pregunta tenga sus encuestados internamente ordenados descendientemente según su valor de opinión, en caso de empate se colocará primero el encuestado con mayor nivel de experticia. También por cada tema, se busca que las preguntas estén ordenadas descendientemente según su promedio del valor de opinión. en caso de empate entre dos o más preguntas, se pondrá primero la pregunta que tiene mayor promedio de experticia, en caso de que persista el empate se pondrá primero aquella que tenga el mayor número de encuestados. Para los temas se busca algo similar, se requiere ordenar los temas de forma descendientemente según el promedio de los promedios de los valores de las opiniones de las preguntas del respectivo tema, en caso de que haya empate entre dos o más temas, el tema con mayor promedio de promedios de experticia se pondrá primero, en caso de persistir el empate se pondrá primero el tema con mayor número de encuestados. Por último, con el fin de saber el nivel de participación de los encuestados en múltiples preguntas de diversos temas, se requiere también generar la lista de todas las personas encuestadas ordenadas descendientemente por su nivel de experticia, en caso de empate se escoge el que tenga el mayor identificador. Se asume que cada persona sólo es consultada una sola vez.

Una instancia del problema sería la siguiente, $K = 2, M = 2, N_{min} = 2$ y $N_{max} = 4$.

Encuestados:

- **1:** Nombre: Sofía García, Experticia: 1, Opinión: 6,
- **2:** Nombre: Alejandro Torres, Experticia: 7, Opinión: 10,
- **3:** Nombre: Valentina Rodríguez, Experticia: 9, Opinión: 0,
- **4:** Nombre: Juan López, Experticia: 10, Opinión: 1,
- **5:** Nombre: Martina Martínez, Experticia: 7, Opinión: 0,
- **6:** Nombre: Sebastián Pérez, Experticia: 8, Opinión: 9,
- **7:** Nombre: Camila Fernández, Experticia: 2, Opinión: 7,
- **8:** Nombre: Mateo González, Experticia: 4, Opinión: 7,
- **9:** Nombre: Isabella Díaz, Experticia: 7, Opinión: 5,
- **10:** Nombre: Daniel Ruiz, Experticia: 2, Opinión: 9,
- **11:** Nombre: Luciana Sánchez, Experticia: 1, Opinión: 7,
- **12:** Nombre: Lucas Vásquez, Experticia: 6, Opinión: 8

Tema 1:

- **Pregunta1.1:** {10, 2}
- **Pregunta1.2:** {1, 9, 12, 6}

Tema 2:

- **Pregunta2.1:** {11, 8, 7}
- **Pregunta2.2:** {3, 4, 5}

La salida para esta instancia seria la siguiente:

[8.25] Tema 1:

[9.50] Pregunta 1.1: (2, 10)

[7.00] Pregunta 1.2: (6, 12, 1, 9)

[3.67] Tema 2:

[7.00] Pregunta 2.2: (8, 7, 11)

[0.33] Pregunta 2.1: (4, 3, 5)

(los valores entre corchetes son los promedios tanto de la pregunta como del tema, es necesario incluirlos en la salida final)

Lista de encuestados:

{4, 3, 6, 9, 5, 2, 12, 10, 7, 11, 1}

Además de esto, la consultoría también necesita obtener algunos datos, específicamente lo siguiente:

- Pregunta con mayor promedio de opiniones.
- Pregunta con menor promedio de opiniones.
- Pregunta con mayor promedio de experticia.
- Pregunta con menor promedio de experticia.
- Encuestado con el mayor nivel de experticia.
- Encuestado con el menor nivel de experticia.
- Encuestado con el menor valor de opinión.
- Encuestado con el mayor valor de opinión.
- Promedio de experticia de los encuestados.
- Promedio del valor de opinión de los encuestados.

El gerente de la consultoría ha escuchado que los estudiantes de Ingeniería de Sistemas, especialmente del curso de ADA I, de la Universidad del Valle, son especialistas en resolver este tipo de problemas y en los menores tiempos posibles y ha decidido contratarlos para resolver su problema.

Aspectos a tener en cuenta (implementación):

Al momento de desarrollar el proyecto de ADA I, debe tener en cuenta los siguientes aspectos:

1. Diferencia de soluciones:

Las 2 soluciones se consideran diferentes siempre y cuando no se reutilicen estructuras de datos, es decir, si se usa por ejemplo un arreglo como estructura de datos para la solución 1, esta no podrá ser reutilizada en la solución 2.

2. Ordenamiento e impresión en la estructura elegida:

Si para una solución se usa una estructura de datos, los datos deben ser cargados, ordenados y mostrados por esa estructura, es decir no es válido usar la estructura de datos de colas, pero previamente haber ordenado los datos en un arreglo, se deben cargar los datos en cola, ordenarlos y mostrarlos mediante sus métodos. En caso de que la estructura elegida no tenga métodos de ordenamiento, sí puede usar una estructura de datos auxiliar, implementando sus métodos.

3. Métodos predefinidos:

Se deben usar métodos propios (implementarlos en código), no es válido usar funciones predeterminadas de Python, como es el caso del método Sort para las listas.

→ Usted debe:

1. Plantear 2 soluciones al problema planteado.
2. Importante: una solución se considera diferente a otra en la medida que utilice diferentes estructuras de datos para almacenar los datos, y por ende en algunos de los algoritmos para manipular los respectivos datos.

→ Para cada una de las soluciones, debe:

- Explicar de manera clara la idea de la solución al problema, qué estructuras de datos va a usar, cuáles algoritmos usará (incluyendo una descripción de ellos).
- Si va a utilizar un algoritmo ya existente, debe indicarlo y explicarlo. Tanto para los algoritmos existentes como los propuestos en el proyecto, debe determinar su complejidad computacional teórica.
- En caso de utilizar un algoritmo ya existente, como, por ejemplo, Merge Sort o cualquier otro, NO es permitido hacer uso de funciones en bibliotecas (librerías) del lenguaje de programación, todos los algoritmos existentes (que se han visto en clase) que vaya a requerir para el proyecto debe escribirlos manualmente, y cerciorarse de que funcionan en la práctica igual que como están planteados en la teoría.

- Set de distintas pruebas con diferentes tamaños y parámetros de entrada (mínimo 4 pruebas de 4 instancias del problema).
- Adjuntar la solución al problema (junto con las instrucciones para ejecutar el código) en uno de los siguientes 3 lenguajes: **java**, **c++**, **python**.
- Se requiere un informe que contenga lo siguiente:
 - Por cada estrategia: explicar de manera clara la idea de la solución al problema, qué estructuras de datos va a usar, cuáles algoritmos usará (incluyendo una descripción algorítmica de ellos).
 - Si va a utilizar un algoritmo ya existente, debe indicarlo y explicarlo. Tanto para los algoritmos existentes como los propuestos en el proyecto, debe determinar su complejidad computacional teórica.
 - En caso de utilizar un algoritmo ya existente, como, por ejemplo, Merge Sort o cualquier otro, NO es permitido hacer uso de funciones en bibliotecas (librerías) del lenguaje de programación, todos los algoritmos existentes (que se han visto en clase) que vaya a requerir para el proyecto debe escribirlos manualmente, y cerciorarse de que funcionan en la práctica igual que como están planteados en la teoría.
 - Análisis de resultados: debe haber comparaciones de la complejidad teórica estimada con la complejidad real del algoritmo en ejecución, comparaciones de tamaño de entrada vs tiempo de salida, esto se hace tomando tiempos de ejecución del algoritmo que soluciona el problema. También, estas comparaciones deben ser presentadas por medio de gráficos de tamaño de entrada vs tiempo de salida, y también debe comparar los tiempos de salida de las distintas soluciones que presente, evidenciando que hay diferencias entre los tiempos de salida de las distintas soluciones que se plantea (que concuerda la mejora en la complejidad computacional teórica).
- Análisis de resultados: 1) se deben hacer comparaciones de la complejidad teórica temporal estimada con la complejidad temporal real de los algoritmos, considerando la relación tamaño de entrada vs tiempo de salida, esto se hace tomando tiempos de ejecución de los algoritmos que solucionan el problema. 2) Estas comparaciones deben ser presentadas por medio de gráficos donde se contrasta tamaño de entrada vs tiempo de salida, 3) se deben hacer comparaciones de los tiempos de salida de las distintas soluciones que se presentan, evidenciando que hay diferencias entre los tiempos de salida de las distintas soluciones que se plantea. 4) Para el análisis se deben utilizar todos los archivos de prueba dados como los que debieron ser propuestos por el grupo.

- Conclusiones del proyecto: se necesita plantear conclusiones claras que reflejan tanto lo descrito en el informe, lo desarrollado en la implementación así como la experiencia adquirida por los estudiantes durante el desarrollo del proyecto.
- Por último, es necesario que los integrantes realicen un video de máximo 15 minutos donde participen todos los integrantes, en el cual se explique brevemente:
 - Idea de las alternativas propuestas.
 - Estructuras de datos a utilizar.
 - Descripción general de los algoritmos más relevantes usados para el ordenamiento y procesamiento de datos para cada solución.
 - Complejidad temporal de los algoritmos más relevantes de ordenamiento, consulta, etc.
 - Lenguaje en el cual se desarrollaron las implementaciones.
 - Breve descripción de la implementación de una de las alternativas.
 - Mostrar un ejemplo de como se usa la implementación de una de las alternativas desarrolladas.

La entrega del video es necesaria y es parte del proyecto. Si no se entrega el video, por una parte, se afecta la nota del proyecto y por otra, dicho grupo tendrá que hacer una sustentación del proyecto con el profesor a coordinar con él.

La entrega se debe realizar por medio del campus virtual en la fecha prevista para ello, por uno sólo de los integrantes del grupo. La fecha de entrega límite es el 12 de diciembre de 2024 a las 23:59.

Se debe entregar un informe en formato pdf, los archivos fuente, un archivo Readme.txt que describa todos los archivos entregados y las instrucciones para ejecutar la aplicación. Todo lo anterior en un solo archivo empaquetado cuyo nombre contiene los apellidos de los autores y cuya extensión corresponde al modo de compresión utilizado.