

Análisis y Diseño de Algoritmos - Proyecto No. 3

Sección 20 - Gabriel Brolo

Mayo 2024

1 Introducción

El objetivo de este proyecto es concretar el concepto cuando el Análisis Amortizado se entrelaza con el Análisis Competitivo y los Algoritmos on-line y de esta forma, se busca que el estudiante practique la implementación de un algoritmo on-line que puede ser estudiado desde la perspectiva del análisis amortizado.

Las condiciones son las siguientes:

- Fecha de entrega: 23 de Mayo de 2024.
- El proyecto es individual.
- Modalidad de entrega: Virtual, sin presentación; video corto y reporte.

2 Instrucciones de Entrega

- Siga las instrucciones de la siguiente sección. No copie el código del algoritmo que se pide realizar (en las referencias se encuentra la propuesta de una implementación), realice su propia implementación. De encontrarse una copia muy similar o exacta del mismo, se colocará una nota de cero puntos. Use la referencia como guía, aprenda de la referencia, pero no copie el código.
- Entregue un reporte breve con las respuestas a las preguntas solicitadas y además entregue un video de 5 minutos (no más, de lo contrario resto puntos) en YouTube y márkelo como no listado, para que de esta forma solo pueda verse al tener el enlace de acceso. En el video, explique brevemente su implementación y muestre las respuestas a las preguntas solicitadas, sin entrar en detalle.
- Trabaje su proyecto en un repositorio privado de GitHub; provea acceso a **gbrolo**.
- En Canvas entregue el PDF de su reporte. En la sección inicial del mismo incluya el enlace a su repositorio privado y el enlace a su video de YouTube. Posteriormente, coloque las respuestas a **todas** las preguntas/actividades planteadas. Respuestas que no salgan de su implementación, no serán aceptadas, debe mostrar en su video como obtiene las respuestas.
- Video: [10 puntos], Repositorio privado con commits: [10 puntos], Reporte en PDF con respuestas: [20 puntos], Implementación del algoritmo: [60 puntos].

3 Instrucciones

Investigue sobre el algoritmo MTF (Move to Front) y realice un programa en el lenguaje de programación de su elección en el que se calculen los costos de acceso totales del algoritmo MTF sobre las siguientes secuencias de solicitudes:

1. Calcular el costo de acceso utilizando el algoritmo MTF para

- (a) Lista de configuración: 0, 1, 2, 3, 4
- (b) Secuencia de solicitudes: 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4

Imprima la lista de configuración, la solicitud, su costo y la configuración de la lista aplicando MTF por cada solicitud en la secuencia y, al final, imprima el costo total de los accesos.

2. Calcular el costo de acceso utilizando el algoritmo MTF para

- (a) Lista de configuración: 0, 1, 2, 3, 4
- (b) Secuencia de solicitudes: 4, 3, 2, 1, 0, 1, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 4

Imprima la lista de configuración, la solicitud, su costo y la configuración de la lista aplicando MTF por cada solicitud en la secuencia y, al final, imprima el costo total de los accesos.

3. ¿Para qué secuencia de 20 solicitudes se obtiene el mínimo costo total de acceso utilizando el algoritmo MTF para la configuración 0, 1, 2, 3, 4? ¿Cuál sería ese costo total de acceso?
4. ¿Para qué secuencia de 20 solicitudes se obtiene el peor de los casos utilizando el algoritmo MTF para la configuración 0, 1, 2, 3, 4? ¿Cuál sería ese costo total de acceso?
5. Calcular el costo de acceso utilizando el algoritmo MTF para

- (a) Lista de configuración: 0, 1, 2, 3, 4
- (b) Secuencia de solicitudes: 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

Imprima la lista de configuración, la solicitud, su costo y la configuración de la lista aplicando MTF por cada secuencia de solicitudes y al final el costo total de acceso. Si se hiciera la secuencia de solicitudes 3, ¿cuál es el costo total de acceso? ¿Se observa algún patrón cuando hay una repetición de 20 elementos en la secuencia?

6. Se plantea un algoritmo mejorado de MTF: **IMTF por Rakesh Mohanty y Sasmitha Tripathy**, basado en el concepto de mirada hacia adelante (look-ahead), en el cual, después de acceder al elemento de la posición i en la lista de configuración, se mueve el elemento al frente de la lista si y sólo si este elemento está en los próximos $i - 1$ elementos del elemento accedido en la solicitud de secuencia. En caso contrario, el elemento accedido no se mueve al frente de la lista de configuración. Imprima la lista de configuración, la solicitud, su costo y la configuración de la lista aplicando IMTF por cada solicitud en la secuencia. Al final, imprima el costo total de acceso usando IMTF para el mejor y el peor de los casos de MTF.

4 Referencias

Puede consultar las siguientes referencias. Recuerde, para la implementación de su código no haga un simple copy-paste del algoritmo, la implementación queda a su creatividad.

- <https://www.geeksforgeeks.org/self-organizing-list-move-front-method/>
- <https://www.cs.princeton.edu/courses/archive/spr09/cos423/Lectures/mtf.pdf>
- <https://arxiv.org/ftp/arxiv/papers/1105/1105.0187.pdf>