

Tarea 2

Análisis amortizado

Profesor: Gonzalo Navarro
Auxiliares: Diego Salas, Asunción Gómez

1. Instrucciones

El objetivo de esta tarea es analizar amortizadamente los costos de búsquedas en dos estructuras: splay trees y árbol rojo-negro. Para esto, deberá implementar ambas estructuras junto a sus operaciones (en el caso del splay tree, todos los zig,zag pertinentes). Luego, deberá insertar los elementos $1, 2, \dots, N$ en ambas estructuras, y sobre una secuencia de elementos, deberá analizar los costos amortizados de búsqueda en cada estructura.

La entrega considera un zip con:

- Código con las estructuras implementadas, los tests y un README para compilar y correr el código.
- Programa que use ambas estructuras y realice los tests solicitados (los inputs que utilizaron, cómo midieron el tiempo, etc).
- README para compilar y correr el código.
- Informe de la tarea.

Las partes de su informe fueron detalladas en el enunciado de la tarea 1. Debe seguir la misma estructura. A continuación, se presenta la distribución de puntaje y qué se espera en cada sección.

2. Introducción (0.5 pts)

En esta sección de su informe, debe presentar el problema y plantear una hipótesis con respecto a los costos esperados de las operaciones y el espacio utilizado.

3. Desarrollo (3.5 pts)

Inserte los elementos $\{1, \dots, N\}$ en su splay tree y en su árbol rojo-negro. Debe probar para $N = \{2^{16}, \dots, 2^{24}\}$.

Luego, deberá crear un arreglo de tamaño $M = 2^{28}$ que representa la secuencia de elementos a buscar.

3.1. Elementos equiprobables (1.5 pts)

En la primera parte, considere que los N elementos tienen la misma probabilidad de ser buscados. Cree un arreglo C de tamaño M que representará su secuencia de búsqueda. Cada elemento debe aparecer la misma cantidad de veces para que sean equiprobables y deben estar desordenados.

Realice una búsqueda de esta secuencia sobre sus estructuras y evalúe empíricamente los costos de las operaciones.

3.2. Elementos con “skew” (2.0 pts)

En esta parte, cada elemento tendrá una probabilidad diferente en la secuencia de búsqueda. Para esto, crearemos 3 datasets distintos de tamaño $M = 2^{28}$ de la siguiente forma:

1. Crear una permutación aleatoria π de $1, \dots, N$.
2. Escribir en un arreglo C , $\phi(i)$ copias del elemento $\pi(i)$.

$$f(i) = \lfloor i^\alpha \rfloor, \alpha = 0.5, 1, 1.5$$

$$SUM = \sum_{i=1}^N f(i) : \text{representa el número total de frecuencias}$$

$$\phi(i) = \frac{f(i)}{SUM} M : \text{generamos frecuencias normalizadas}$$

3. Reordenar aleatoriamente el arreglo C para que no vengan juntos los mismos números. Fijarse que C tiene tamaño M .

Como pueden ver, $\alpha = 0.5$ nos dará el primer dataset, $\alpha = 1$ el segundo y $\alpha = 1.5$ el tercero.

Igual que en la parte anterior, realice una búsqueda de cada dataset sobre su splay tree y su árbol rojo-negro. Evalúe los costos de sus operaciones.

4. Resultados (1 pto)

Presente en su informe gráficos con sus tiempos para ambos tipos de secuencia: equiprobables y con “skew”. En el primer caso, realice 5 tests con distintas secuencias de búsqueda de tamaño M por cada $N=2^{16}, \dots, 2^{24}$. Calcule el promedio de los 5 tests realizados para cada (N) , su varianza y desviación estándar. En el segundo caso, deberá realizar 5 secuencias distintas para cada α . Calcule el promedio de los 5 tests realizados para cada par (N, α) . Calcule su varianza y desviación estándar.

5. Conclusiones (1 pto)

Analice y comente sus resultados, para luego concluir. Debe corroborar (o no) su hipótesis planteada anteriormente. Comente si sus resultados siguen la teoría y analice porque sí o no.

6. Especificaciones

- La tarea se entregará vía U-Cursos.
- La tarea puede realizarse en grupos de a 3. **Al menos** uno de los integrantes debe ejecutar los tests y mencionar en el informe en qué computadores se corrieron los algoritmos. Considere los tamaños de sus memorias caché y RAM.
- Deben seguir las buenas prácticas a la hora de implementar su código. Un programa comentado ayuda a que se pueda revisar mejor.
- Se recomienda utilizar C, C++ o Java para el código y \LaTeX para su informe.
- Recuerde que en su código debe añadir un README (no se pidió en la tarea anterior).