

# Proyecto 1: Skip Lists

## Estructuras de Datos, Primer Semestre 2021

Prof. Diego Seco

Ayudantes: Alexis Espinoza, Catalina Pezo, Alexander Irribarra, Lucas Kraemer

### 1. Descripción del Problema

Una *Skip List* es una estructura de datos estrechamente relacionada con las vistas en esta asignatura – en concreto con listas enlazadas y con el tema de Diccionarios – pero que no nos alcanza el tiempo para verla en clases de teoría. Para pagar la deuda histórica de la asignatura con las marginadas Skip Lists, este año se implementarán como primer proyecto de la asignatura.

El primer paso de este proyecto consiste en leer acerca de esta estructura de datos. Para ello, se recomiendan las slides del libro de referencia de la asignatura, aunque es una estructura de datos básica cubierta por la mayoría de libros de estructuras de datos. También están descritas en <https://www.geeksforgeeks.org/skip-list> (no se puede copiar el código fuente de este recurso ni de ningún otro.)

Una vez leída la información, hay dos primeras conclusiones evidentes: i) tienen mucho en común con las listas enlazadas y ii) su análisis es probabilístico. En cuanto a este segundo punto, dado que análisis probabilístico es contenido de la asignatura de Análisis de Algoritmos, no será objetivo de este proyecto. Es decir, el análisis que se pedirá seguirá siendo de peor caso y también se hará un análisis experimental.

### 2. Implementación

Se implementará un Tipo Abstracto de Datos, que llamaremos MiniSet, representando un conjunto de enteros y soportando las siguientes operaciones: `insert(int)`, `remove(int)`, `search(int)`. La operación `search` devolverá verdadero si el valor ha sido previamente insertado en el conjunto y falso en caso contrario. La operación `insert`, insertará al elemento en el conjunto (si ya existe en el conjunto, no hará nada). La operación `remove`, borrará al elemento del conjunto en caso de que exista.

Se implementarán dos estructuras de datos para MiniSet, una basada en listas enlazadas y otra basada en Skip Lists.

Para la implementación basada en listas enlazadas, debe tomar la decisión de mantener la lista ordenada o no. Ambas opciones son válidas. En el informe debe indicar claramente qué decisión tomó. Además, debe realizar un análisis teórico de peor caso de la operación `search`. Incluya también un análisis del espacio que ocupará la estructura. Ver en el siguiente párrafo las condiciones que se considerarán.

Para la implementación basada en Skip Lists, también debe realizar un análisis teórico de peor caso de las 3 operaciones implementadas. Importante: el análisis debe ser de peor caso y NO probabilístico. Incluya también un análisis del espacio que ocupará la estructura en peor caso. Además, en el informe debe indicar cualquier decisión de implementación que haya tomado. En particular, debe definir el número máximo de niveles a emplear y la probabilidad con la que cada elemento de un nivel  $i$  se almacena también en el nivel

$i + 1$ . Existen análisis matemáticos que indican cómo se relacionan ambos parámetros, por simplicidad, para la evaluación experimental puede considerarse un máximo de 4 niveles y una probabilidad de  $1/2$ . Para el análisis teórico considere  $\log n$  niveles y que el nivel  $i$  tiene exactamente la mitad de elementos que el nivel  $i - 1$  (el nivel base es el 0 con  $n$  elementos, el nivel 1 tiene  $n/2$ , etc.).

Además, se compararán experimentalmente ambas soluciones. Es decir, el informe debe incluir 3 gráficas, una por cada operación, donde el eje X represente número de elementos y el eje Y represente tiempo de ejecución de 1 operación de ese tipo. Para generar la secuencia de valores a insertar, se creará un vector de valores de 1 a  $n$ , el cual será reordenado (en C++ se puede hacer con [http://www.cplusplus.com/reference/algorithm/random\\_shuffle/](http://www.cplusplus.com/reference/algorithm/random_shuffle/), existiendo alternativas equivalentes en otros lenguajes).

En el informe también se deben incorporar conclusiones acerca de lo predicho por la teoría y lo obtenido en la experimentación. En concreto, debe ser capaz de justificar una respuesta a la pregunta ¿qué implementación es mejor?

### 3. Evaluación

El proyecto se realizará en parejas. Enviar en Canvas lo siguiente:

1. Un informe que:
  - a) Incluya portada, descripción de la tarea, descripción de las soluciones propuestas, detalles de implementación, análisis teórico y análisis experimental.
  - b) Sea claro y esté bien escrito. Un informe difícil de entender es un informe que será mal evaluado aunque todo esté bien implementado. La persona que revise el documento debe poder entender su solución sólo mirando el informe.
  - c) Esté en formato pdf.
2. Un archivo comprimido con todos los ficheros fuente implementados para solucionar la tarea. El informe debe hacer referencia a ellos y explicar en qué consiste cada uno.

**Fecha de Entrega: viernes 28 de mayo 11:59PM**