

Proyecto 1  
Estructuras de Datos, Primer Semestre 2017  
Prof. Diego Seco  
Ayudantes: Diego Gatica y Paulo Olivares

## 1. Ejercicios

1. [1 punto] Cree una pequeña librería (entendida como conjunto de clases) que contenga los ADT List y Stack junto con algunas implementaciones de las mismas realizadas en los laboratorios (ArrayList y LinkedList para List) o vistas en clase (Stack basado en la LinkedList anterior). Adjunte un fichero test\_lib.cpp que permita probar la funcionalidad de la librería. Documente bien esta librería ya que se podrá requerir más adelante en futuros laboratorios y mini-proyectos. Específicamente, se debe incluir un diagrama de clases UML y se debe indicar la complejidad de cada operación en notación asintótica.
2. [2 puntos] Además de los algoritmos de búsqueda lineal y búsqueda binaria vistos en el laboratorio 2, existen otros algoritmos de búsqueda como la búsqueda exponencial (también conocida como doblada o galopante). Puede encontrar la descripción de la misma en [https://en.wikipedia.org/wiki/Exponential\\_search](https://en.wikipedia.org/wiki/Exponential_search). En este ejercicio se le pide implementar dicho algoritmo y compararlo con los desarrollados en el laboratorio 2. Todos los algoritmos se deben adaptar para trabajar sobre el ArrayList implementado en el ejercicio anterior. La comparación se hará experimentalmente y deberá entregar los siguientes gráficos de tiempo de ejecución de los algoritmos para diferentes tamaños de entrada.
  - a) Se tomarán tiempos de búsqueda de elementos aleatorios.
  - b) Los elementos buscados siempre se encuentran entre los primeros (10%).
  - c) Los elementos buscados siempre son mayores que el mayor de los elementos.

Comente acerca de los resultados obtenidos.

3. [3 punto] Dadas dos listas ordenadas, una operación interesante es aquella que computa la intersección de las mismas y la retorna en una nueva lista también ordenada. Por ejemplo, dadas las listas  $A = 3- > 7- > 10$  y  $B = 1- > 7- > 10- > 12- > 20$ ; la lista resultado sería  $C = 7- > 10$ . En este ejercicio se implementarán dos versiones de dicha operación empleando el ADT List del ejercicio 1. La selección de la estructura de datos más adecuada para cada versión debe ser justificada.
  - Versión 1: se irán recorriendo las dos listas de forma sincronizada, copiando el menor de los elementos a la lista resultado y avanzando al siguiente elemento de la lista donde se encuentra dicho elemento (mientras que en la otra lista no se avanza).
  - Versión 2: cada elemento de la lista más corta se buscará, mediante el algoritmo de búsqueda binaria, en la lista más larga. En caso de encontrarse, se añadirá a la lista resultado. En caso contrario, se omitirá y se pasará al siguiente elemento. (Si las dos listas tienen igual longitud, se tomará cualquiera de ellas como la menor).

En este ejercicio, se debe entregar, además del código, una evaluación teórica y experimental de los dos algoritmos. Para la comparación teórica, asuma que  $n$  es el tamaño de la lista más corta y  $m$  es

el tamaño de la lista más larga. Para la comparación experimental, se harán diferentes experimentos empleando listas generadas aleatoriamente con distintos tamaños. En un primer experimento se asumirá que ambas listas son de igual tamaño,  $n = m$ , y se graficará el tiempo que toman dichos algoritmos a medida que se aumenta el número de elementos. En un segundo experimento se variará la razón  $n/m$  (por ejemplo,  $n/m = 1/2$  significa que la lista más larga es del doble de tamaño que la más corta,  $n/m = 1/4$  es cuatro veces más grande, etc.). El tamaño de la lista más corta será 100,000. Se debe comentar acerca de los resultados obtenidos.

## 2. Evaluación

El proyecto se realizará en parejas. Enviar en piazza (mensaje privado a *Instructors*) lo siguiente:

1. Un informe que:
  - a) Incluya portada y respuestas a todas las preguntas de la tarea.
  - 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 21 de Abril 11:59PM**