



Estructuras de Datos y Algoritmos Avanzados (2022-2) Proyecto 2: Conteo de patrones

Profesor: José Fuentes Sepúlveda

Ayudante: Alexander Iribarra Cortés

Objetivos

Los objetivos de este proyecto son:

- Mejorar la programación, compilación y ejecución de programas escritos en lenguaje *C++* u otros.
- Estudiar e implementar algoritmos y estructuras de datos para la búsqueda de patrones en texto.
- Analizar experimental y teóricamente el desempeño de los algoritmos y estructuras de datos implementadas.

1. Descripción del problema

La búsqueda de patrones en texto corresponde a uno de los problemas fundamentales en Ciencias de la Computación y vamos su aplicación cotidianamente. Por ejemplo, cada vez que presionamos la combinación de teclas **control + F** o **control + S**, a nivel de implementación se está utilizando una estructura de datos especializada para búsqueda en texto.

En este proyecto se implementarán tres soluciones para soportar búsqueda de patrones:

1. Algoritmo de búsqueda: Implementación de uno de los siguientes algoritmos: Boyer-Moore, Boyer-Moore-Horspool, Knuth-Morris-Prat o Karp-Robin. (Ver clase 3)
2. Búsqueda basada en arreglo de sufijos: Utilizando como base el arreglo de sufijos de un texto, implementar la búsqueda de patrones usando búsqueda binaria. (Ver clase 3)

3. FM-index: Implementación simple del auto-índice FM-index, usando la Burrows-Wheeler Transform, los arreglos *Occ* y *C*, y la función LF-mapping. (Ver clase 6)

2. Objetivos específicos

La entrega del proyecto, que consiste de un informe y el código fuente, debe satisfacer los siguientes objetivos:

- Cada estructura de datos y algoritmo deben proveer de la funcionalidad $count(T, p)$, la cual retorna la cantidad de veces que el patrón p aparece en el texto T
- Describir las características principales de los algoritmos y estructuras de datos implementadas.
- Describir las decisiones de implementación más importantes.
- Plantear varias hipótesis sobre el rendimiento de las soluciones a comparar en escenarios específicos. Por ejemplo, “La solución X es más rápida buscando patrones que la solución Y cuando los datos cumplen cierta característica C ” o “la solución X usa menos espacio que la solución Y ”. Se deben plantear (por lo menos) tantas hipótesis como integrantes tenga el equipo.
- Diseñar un experimento que permita verificar cada hipótesis.
- Ejecutar los experimentos y discutir los resultados obtenidos.

3. Condiciones

- El proyecto se realizará en grupos de dos o tres estudiantes. El informe debe reflejar claramente los autores del proyecto.
- Las soluciones a través de algoritmos y estructuras de datos pueden ser obtenidas de una fuente externa, la cual debe ser referenciada en el informe. Además, el funcionamiento de las implementaciones obtenidas externamente deben estar detalladamente comentadas.
- Todas las implementaciones deben realizarse en el mismo lenguaje de programación y utilizar los mismos tipos de optimizaciones.
- Las implementaciones se pueden realizar en los lenguajes *C++*, *Java* o *Python*.
- Todos los experimentos deben ser llevados a cabo en la misma máquina.

- En el informe se debe describir el entorno de *hardware* y *software* en que se ejecutan los experimentos. Además, debe quedar clara la manera en que se generan u obtienen tanto los datos de entrada como los resultados reportados. Se pueden obtener los datos de una fuente externa, como puede ser el corpus de *Pizza&Chili*¹. Una persona externa con la información descrita en el informe y el código fuente, debiera ser capaz de replicar los experimentos y obtener las mismas conclusiones.
- Con el objetivo de reducir ruido en los resultados, las mediciones realizadas deben contemplar al menos 30 repeticiones, de las cuales se reportará la media y varianza.

4. Evaluación

El proyecto se realizará en grupos de dos o tres personas. Se debe subir a 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, considerando las condiciones previamente estipuladas.
 - b) Sea claro y esté bien escrito. Un informe difícil de entender será mal evaluado, aunque todo esté bien implementado. Quien revise el documento debe poder entender su solución solo 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 4 de noviembre de 2022 11:59PM

¹<http://pizzachili.dcc.uchile.cl/>