

Tarea 1: Esquemas de Decodificación Instantánea - INFO145

Diseño y Análisis de Algoritmos.

Académico: Héctor Ferrada.
Instituto de Informática, Universidad Austral de Chile.

Junio 13, 2020

Entrega. Debe subir todo a siveduc en un archivo, t1Apellidos.zip (3 integrantes por grupo), con su implementación (incluya lo necesario para su ejecución, como Makefile en el caso de C++) y un informe claro y conciso. Fecha de entrega: **Lunes 06 de Julio.**

Informe. Este debe explicar la lógica que uso en sus implementaciones y la justificación de ello, p.ej. explicar porqué no uso otra alternativa más eficiente si la hubiese. Su informe al menos debe contemplar: introducción, metodología, experimentación y conclusiones.

Rendimiento en Algoritmos de Codificación/Decodificación

El objetivo de este trabajo es adquirir un mayor dominio en el análisis de algoritmos y de estructuras de datos para resolver un mismo problema. Se evalúa que usted pueda llevar a la práctica la teoría vista en clases, implementando los algoritmos de **Shannon Fano** y de **Huffman** para codificar símbolos, chequeando el performance empírico de sus soluciones implementadas.

Codificación de secuencias de símbolos. El problema a resolver es el siguiente. Dado un texto $T[1..u]$, construido sobre un alfabeto $\Sigma[1..\sigma]$, se desea construir un bitstream (secuencia de bits) $B[1..n]$, por medio de los esquemas de decodificación instantánea ya introducidos, que represente unívocamente a T . Deberá estimar el espacio exacto en RAM que requieren sus estructuras construidas y ejecutar experimentos que midan la velocidad de decodificación de sus implementaciones a fin de incluir un buen análisis.

Algoritmo de Shannon Fano. Corresponde al mismo algoritmo estudiado en clases, el cual deberá implementar para construir su representación $F[1..n]$. Debe además **proponer** un algoritmo de decodificación, explique su nuevo algoritmo y entregue el análisis del tiempo de ejecución en función del tamaño de la entrada n y de σ si es necesario.

Algoritmo de Huffman Canónico. Corresponde al mismo algoritmo estudiado en clases, el cual deberá implementar para construir su representación $H[1..m]$; además de el algoritmo de decodificación que también fue entregado en clases.

Al construir las estructuras debe considerar que el espacio total de cada método debe incluir al bitstream junto a todas las estructuras de datos necesarias para ejecutar la decodificación. Para el caso de cada bitstream $F[1..n]$ y $H[1..m]$, puede utilizar arreglos booleanos para representar los bits. Considere para sus análisis que, en el modelo RAM, se considera tiempo constante la lectura/escritura de $O(\log n)$ bits; por ejemplo, esto se ve en el algoritmo de decodificación de Huffman canónico al formar el entero N en

un solo paso; a pesar de que, quizá, en su implementación lo haga bit a bit y no los $h = O(\log n)$ bits de un solo paso (esto si no dispone de alguna librería adicional que le permita leer/excribir h bits desde/en un entero).

Experimentación. Descargue los siguientes archivos para ejecutar sus experimentos:

- english.100MB.
- dna.100MB.

Utilice estos archivos para ejecutar experimentos con texto en inglés y con secuencias de ADN. Cada uno de estos archivos hace el papel del $T[1..u]$ de entrada. Para cada T , construya sus estructuras codificadas F y H , luego realice al menos los siguientes dos experimentos —desde luego, puede agregar más experimentos si estima que es necesario para su análisis y presentación clara de los resultados.

1. **Tiempo de extracción de prefijos del texto.** Para diferentes posiciones aleatorias del texto $k \leq u$, extraiga $T[1..k]$ desde sus representaciones codificadas.
2. **Tiempo de extracción de segmentos aleatorios del texto.** Previó a la ejecución de experimento, **deberá proponer nuevos algoritmos** que le permitan recuperar **eficientemente** cualquier segmento $T[i..j]$, con $1 < i \leq j \leq u$, sin la necesidad de comenzar la decodificación desde el inicio de su representación codificada (evitando hacer lo que se hizo en el experimento anterior). Explique sus nuevos algoritmos así como las estructuras de datos adicionales incluidas para poder llevarla acabo su nueva metodología. Considere en sus gráficos el espacio adicional que las nuevas estructuras requieren.

Presente los resultados de cada experimento gráficamente, a fin de poder luego analizar y concluir sobre los eventos.