Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2283 - Diseño y Análisis de Algoritmos
Profesor: Nicolás Van Sint Jan
Ayudante: Dante Pinto

Ayudantía 5

Algoritmos Codiciosos

Problema 1

Demuestre que dada una función de frecuencias relativas $f : \Sigma \to (0, 1)$, el algoritmo de Huffman encuentra una Σ-codificación, τ, libre de prefijos que minimiza la función $lp_f(\tau)$.

Problema 2

Sean A un arreglos de números naturales de largo n. Definimos la operación $\texttt{Cmp}(A, i, j)$, con $0 \leq i < j \leq n$, como la compresión de elementos del arreglo, que consiste en tomar el subarreglo $A_{ij} = [a_i, a_{i+1}, ..., a_j]$, sumar sus elementos y luego reemplazar el subarreglo, dentro de A, por esta suma.

Por ejemplo, sea $A = [1, 10, 100, 1000, 10000]$, tendremos que $\texttt{Cmp}(A, 1, 3) = [1, 1110, 10000]$, $\texttt{Cmp}(A, 0, 1) = [11, 100, 1000, 10000]$ y $\texttt{Cmp}(A, 2, 4) = [1, 10, 11100]$.

Dado dos arreglos A y B, queremos averiguar si es posible comprimir los arreglos hasta que ambos sean iguales[1] y, si es el caso, encontrar el largo de máximo para el que esto es posible.

Entregue un algoritmo que dado dos arreglos A y B, de largos n y m respectivamente, retorne el largo máximo para el que los arreglos son iguales luego de comprimirlos una cantidad arbitraria de veces. Analice la complejidad del algoritmo.

Problema 3 (Propuesto)

Para el año 2023 la Universidad decidió cambiar el sistema banner y, más aún, el sistema de módulos que tenemos actualmente. En el nuevo sistema, cada profesor decide de forma independiente tanto el horario como la extensión de sus clases y cada alumno debe decidir según eso qué cursos desea tomar.

Para planear tu semestre, armas una lista de cursos para los que cumples los prerequisitos y, como buen estudiante de ingeniería, te propones tomar la mayor cantidad de cursos posibles para el próximo semestre, pero evitando topes de horarios.

Dada una lista de cursos c_1, c_2, ..., c_n, donde cada c_i está dado por una tupla (s_i, e_i), tal que s_i corresponde al inicio del módulo y e_i a su término, buscamos un algoritmo que maximice la cantidad de cursos que pueden tomarse sin superposición.

[1] Decimos que dos arreglos son iguales si tienen el mismo largo y son iguales elemento a elemento.