

Auxiliar 4 - Recurrencia y Programación Dinámica

Profesores: Nelson Baloian
Jeremy Barbay
Patricio Poblete

Auxiliares: Gabriel Flores, Sven Reisenegger
Juglar Díaz, Gabriel Norambuena
Cristóbal Muñoz

P1. Stooge Sort

Stooge sort es un algoritmo muy poco eficiente para ordenar una secuencia de números. El objetivo de este problema es analizar qué tan malo es el algoritmo. Este algoritmo funciona de la siguiente manera:

- (a) Si el valor en el índice 0 es mayor que el valor en el último índice, entonces intercambia ambos valores.
- (b) Recursivamente:
 - i. Ordena los primeros $2/3$ del arreglo con stoogeSort
 - ii. Ordena los últimos $2/3$ del arreglo con stoogeSort
 - iii. Ordena los primeros $2/3$ del arreglo con stoogeSort

Escriba la ecuación de recurrencia para este algoritmo y luego resuélvala.

P2. Algoritmo de Strassen

Para obtener el producto P de dos matrices A y $B \in \mathcal{M}_{n \times n}$ se pensaba que la única forma era calcular $p_{ij} = \sum_k^n a_{ik} b_{kj}$, como seguro recordará de su curso de Álgebra Lineal.

- (a) Formule la ecuación de recurrencia. *El orden del algoritmo es de la forma $O(n^k)$.*

Si se toma n como potencia de 2, y se dividen las matrices en cuadrantes, se puede demostrar (al menos, Strassen pudo) que hay un conjunto de 7 elementos ($\{M_1, \dots, M_7\}$) formados por productos de combinaciones de los 8 cuadrantes de A y B . Estos elementos forman una base a partir de la cual se pueden obtener los cuadrantes del resultado P . Los términos son:

$$\begin{array}{ll} M_1 : & (A_{11} + A_{22})(B_{11} + B_{22}) \\ M_2 : & (A_{21} + A_{22})B_{11} \\ M_3 : & A_{11}(B_{12} - B_{22}) \\ M_4 : & A_{22}(B_{21} - B_{11}) \\ M_5 : & (A_{11} + A_{12})B_{22} \\ M_6 : & (A_{21} - A_{11})(B_{11} + B_{12}) \\ M_7 : & (A_{12} - A_{22})(B_{21} + B_{22}) \end{array}$$

- (b) Considerando que en cada término M_i , el producto puede descomponerse de la misma manera, y considerando que el costo de la suma de matrices cuadradas de lado n es $O(n^2)$, obtenga la ecuación de recurrencia para el Algoritmo de Strassen. Para calcular los cuadrantes de P , sólo basta hacer 8 sumas de los términos de $\{M_1, \dots, M_7\}$. ¿Cuál es una cota superior para éste?

P3. Distancia de Edición

Dados dos strings S y T queremos encontrar la mínima cantidad de pasos para transformar S en T . Para esto contamos con 4 posibles acciones para cada letra:

- Insertar
- Eliminar
- Cambiar
- Mantener

Todas las opciones, excepto mantener una letra, se consideran 1 paso. Por ejemplo, si queremos transformar el String *casa* a *calle*, tenemos que hacer 3 operaciones: Primero cambiamos la letra *s* en *l*, nos queda *cala*, luego cambiamos la última letra *a* en *e*, quedando *cale* y finalmente agregamos una *l* y llegamos a *calle*.

Programa un método `public static int distanciaEdicion(String S, String T)` que basado en una estrategia de Programación Dinámica, resuelva la distancia de edición entre Strings.

P4. BONUS

Obtenga las cotas superiores para los siguientes problemas utilizando el teorema maestro.

- (a) $T(n) = 2 T(\frac{n}{2}) + n^3$
- (b) $T(n) = 16 T(\frac{n}{4}) + n^2$
- (c) $T(n) = 7 T(\frac{n}{3}) + n^2$
- (d) $T(n) = 7 T(\frac{n}{2}) + n^2$
- (e) $T(n) = 2 T(\frac{n}{4}) + \sqrt{n}$
- (f) $T(n) = 3 T(\frac{n}{2}) + n \log(n)$
- (g) $T(n) = 4 T(\frac{n}{2}) + n^2 \sqrt{n}$
- (h) $T(n) = T(\frac{9n}{10}) + n$
- (i) $T(n) = 2 T(\frac{n}{2}) + n \log(n)$