

Estructuras de Datos - Lista de Ejercicios

Prof. José M. Saavedra R.

25 de enero de 2009

1. Dada una cadena, guardar en un arreglo todos sus sufijos y reportarlos ordenados de menor a mayor.
2. Obtener el sufijo propio más largo de una cadena (Nota: puede usar la solución anterior), donde el largo de una cadena se define como:

$$largo(cad) = \sum_{i=0}^{longitud(cad)-1} cod(c(i, cad))$$

donde $c(i, cad)$ es el caracter de las posición i de cad y $cod(x)$ es un código para un caracter x que se define como $cod(x) = ascii(x) - 97$.

Generalice su implementación para cualquier arreglo de cadenas.

3. Suponiendo que su alfabeto solamente permite los caracteres 'a'...'z', implemente un programa que permita determinar la frecuencia de cada caracter en un texto ingresado por consola.
4. Generar una lista de las 10 palabras más comunes que existen en un texto almacenado en un archivo.
5. Torneo: Supongamos que tenemos un arreglo de enteros. El algoritmo de torneo consiste en obtener el máximo o mínimo de los valores, comparándolos en pares. Cada comparación devuelve un ganador que luego será comparado con otro de los ganadores de algún par en la siguiente ronda. El ganador de la última ronda será el máximo o mínimo del arreglo. Lo interesante de este algoritmo es que se puede encontrar el segundo máximo o mínimo solamente comparando los que perdieron contra el ganador del torneo. Implemente un programa que devuelva el máximo y segundo máximo de un arreglo de enteros aplicando el algoritmo del torneo.
6. Se tiene un arreglo que almacena palabras, como si fuera un diccionario. Implemente un programa que reciba como entrada una palabra y reporte si dicha palabra existe en el diccionario, si no existe reporte la más parecida. La más parecida es aquella que tiene más coincidencias evaluando caracter a caracter.
7. Siguiendo el problema anterior, ahora suponga que el diccionario se encuentra almacenado en un archivo de texto. Cada palabra se almacena en una línea del texto.
8. Piense en cómo implementaría un diccionario en donde cada palabra tiene asociada una definición.
9. Una forma muy utilizada de evaluar la similitud entre cadenas es la distancia de edición. La distancia de edición entre dos cadenas es el mínimo número de operaciones (insertar, eliminar o reemplazar) necesarias para que una cadena sea igual a la otra. La forma recursiva es:

$$D(s1_n, s2_m) = \begin{cases} \text{Si} & s1[n] = s2[m] \Rightarrow D(s1_{n-1}, s2_{m-1}) \\ \text{sino} & \Rightarrow \min(D(s1_{n-1}, s2_m), D(s1_n, s2_{m-1}), D(s1_{n-1}, s2_{m-1})) + 1 \end{cases}$$

donde $s1_n$ representa la primera cadena de longitud n y $s2_m$ es la segunda cadena de longitud m .

Implemente un programa que calcule la distancia de edición entre dos cadenas. Piense cómo puede usar una matriz para realizar este cálculo.

10. Se tiene una matriz de n filas por m columnas, el usuario debe ingresar un número entero entre 1 y $n \times m$, si el valor ingresado sale del rango entonces se emitirá un error y el usuario debe ingresar otro valor. Los números se van almacenando correspondientemente en cada casillero de la matriz (Ej, si $n = 3$ y $m = 4$, el número 10 debe ocupar la cuarta fila, primera columna), la matriz solamente tiene capacidad para los $n \times m$ números permitidos. Cada vez que se ingresan números repetidos válidos, el programa emite un mensaje *número ya existe* y se registra dicha ocurrencia.

Implementar un programa que resuelva el problema anterior, el programa debe permitir ingresar iterativamente cada número, siempre que el usuario lo desee. En caso de no querer ingresar más números se mostrará la matriz con los números ingresados así como un reporte de la cantidad de veces que se ingresó cada número.