

Algoritmos y Estructuras de Datos II - 1° cuatrimestre 2019
Práctico 2 - Parte 1

ejercicios para el lunes 1/4

1. Escribir un algoritmo que dada una matriz a : **array**[1.. n ,1.. m] **of int** calcule el elemento mínimo. Escribir otro algoritmo que devuelva un arreglo **array**[1.. n] con el mínimo de cada fila de la matriz a .
2. El arreglo med :**array**[1980..2016,enero..diciembre,1..28,Temp..Prec] **of nat** es un arreglo multidimensional que contiene todas las mediciones estadísticas del clima para la ciudad de Córdoba desde el 1/1/1980 hasta el 28/12/2016. Se asumen definidos los tipos enumerados (enero, febrero, ..., diciembre) y (Temp, TempMax, TempMin, Pres, Hum, Prec). Por ejemplo, $med[2014,febrero,3,Pres]$ indica la presión atmosférica que se registró el día 3 de febrero de 2014. Todas las mediciones están expresadas con números enteros. Por simplicidad asumiremos que todos los meses tienen 28 días.
 - (a) Dar un algoritmo que obtenga la menor temperatura mínima (TempMin) histórica registrada en la ciudad de Córdoba según los datos del arreglo.
 - (b) Dar un algoritmo que devuelva un arreglo que registre para cada año entre 1980 y 2016 la mayor temperatura máxima (TempMax) registrada durante ese año.
 - (c) Dar un algoritmo que devuelva un arreglo que registre para cada año entre 1980 y 2016 el mes de ese año en que se registró la mayor cantidad mensual de precipitaciones (Prec).
 - (d) Dar un algoritmo que utilice el arreglo devuelto en el inciso anterior (además de med) para obtener el año en que ese máximo mensual de precipitaciones fue mínimo (comparado con los de otros años).
 - (e) Dar un algoritmo que obtenga el mismo resultado sin utilizar el del inciso (c).
3. Dado el tipo

```
type person = tuple
    name: string
    age: nat
    weight: nat
end tuple
```

- (a) escribí un algoritmo que calcule la edad y peso promedio de un arreglo a : **array**[1.. n] **of person**.
- (b) escribí un algoritmo que ordene alfabéticamente dicho arreglo.

ejercicios para el miércoles 3/4

4. Dados dos punteros p, q : **pointer to int**
 - (a) escribí un algoritmo que intercambie los valores referidos sin modificar los valores de p y q .
 - (b) escribí otro algoritmo que intercambie los valores de los punteros.Sea un tercer puntero r : **pointer to int** que inicialmente es igual a p , y asumiendo que inicialmente $*p = 5$ y $*q = -4$ ¿cuáles serían los valores de $*p$, $*q$ y $*r$ luego de ejecutar el algoritmo en cada uno de los dos casos?
5. Escribí un algoritmo que devuelva el número (la cantidad) de nodos de una lista enlazada.
6. Escribí un algoritmo que devuelva el mínimo de una lista enlazada de enteros.
7. Escribí un algoritmo que agregue un nodo al final de una lista enlazada no vacía.
8. Escribí un algoritmo que construya una lista enlazada de números naturales con n nodos, todos ellos con el valor 0.
9. Escribí un algoritmo que libere toda la memoria utilizada por una lista enlazada.

ejercicios adicionales

10. Dados dos arreglos $a, b : \mathbf{array}[1..n] \text{ of nat}$ se dice que a es “lexicográficamente menor” que b si existe $k \in \{1 \dots n\}$ tal que $a[k] < b[k]$, y para todo $i \in \{1 \dots k - 1\}$ se cumple $a[i] = b[i]$. En otras palabras, si en la primera posición en que a y b difieren, el valor de a es menor que el de b . También se dice que a es “lexicográficamente menor o igual” a b si a es lexicográficamente menor que b o a es igual a b .
 - (a) Escribir un algoritmo `lex_less` que recibe ambos arreglos y determina si a es lexicográficamente menor que b .
 - (b) Escribir un algoritmo `lex_less_or_equal` que recibe ambos arreglos y determina si a es lexicográficamente menor o igual a b .
 - (c) Escribir un algoritmo `lex_compare` que recibe ambos arreglos y devuelve valores en el tipo enumerado (menor, igual, mayor). ¿Cuál es el interés de escribir este algoritmo?
11. Escribir un algoritmo que dadas dos matrices $a, b : \mathbf{array}[1..n, 1..m] \text{ of nat}$ devuelva su suma.
12. Escribir un algoritmo que dadas dos matrices $a : \mathbf{array}[1..n, 1..m] \text{ of nat}$ y $b : \mathbf{array}[1..m, 1..p] \text{ of nat}$ devuelva su producto.
13. Dado un arreglo `cal2016` : $\mathbf{array}[1..52, \text{mon..fri}, 9..17] \text{ of string}$, donde el primer índice refiere a las semanas del año 2016, el segundo a los días hábiles de la semana y el tercero a las horas laborales del día, dar un algoritmo que devuelve la misma información en un arreglo con el formato modificado: $\mathbf{array}[9..17, 1..52, \text{mon..fri}] \text{ of string}$.
14. Dado un arreglo `cal` : $\mathbf{array}[2010..2020, 1..52, \text{mon..fri}, 9..17] \text{ of string}$, escribir un algoritmo que devuelva el número de horas en las que el calendario no señala ninguna actividad (la celda contiene el string vacío).