Algoritmos y Estructuras de Datos II - 1° cuatrimestre 2020

Práctico 2 - Parte 1

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. Dados los tipos enumerados

```
type mes = enumerate
             enero
             febrero
             ...
             diciembre
           end enumerate
```

```
type clima = enumerate
               Temp
               TempMax
               TempMin
               Pres
               Hum
               Prec
             end enumerate
```

   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. 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.

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. Dados dos arreglos a, b : **array**[1..n] **of nat** se dice que a es "lexicográficamente menor" que b sii 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 sii 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?

6. Escribir un algoritmo que dadas dos matrices a, b: **array**[1..n,1..m] **of nat** devuelva su suma.

7. Escribir un algoritmo que dadas dos matrices a: **array**[1..n,1..m] **of nat** y b: **array**[1..m,1..p] **of nat** devuelva su producto.