

## 2<sup>do</sup> Parcial - 22 de Noviembre de 2012

|                   |                     |  |
|-------------------|---------------------|--|
| <b>Problema 1</b> | 12 (2,2,2,3,3) ptos |  |
|-------------------|---------------------|--|

- a) Las matrices dispersas se utilizan para representar matrices completas que contienen muchos ceros y cuyos valores distintos de cero se encuentran dispersos en la matriz. Dicha representación busca reducir el espacio requerido para almacenar la matriz en la memoria de la computadora y acelerar los cálculos. Para ello se evita almacenar los ceros que estarían presentes en la matriz completa, indicando solamente en que posiciones se encuentran los valores distintos de cero.
- b) Formato elemental: Utiliza tres vectores para representar los elementos distintos de cero. Para cada elemento distinto de cero de la matriz completa se almacena un elemento en cada vector. El primer vector contiene los datos (vector de datos), es decir los valores distintos de cero, el segundo vector contiene las filas en las que dichos elementos se encuentran en la matriz (vector de filas), y el tercer vector almacena las columnas que ocupan los elementos en la matriz (vector de columnas).

Formato Csr, formato Comprimido de almacenamiento por fila: También se utilizan tres vectores para almacenar la matriz. En el vector de datos se almacenan los valores de los elementos de la matriz que son distintos de cero. En el segundo vector se almacenan las columnas donde aparecen cada uno de estos elementos y por último en el tercer vector (de tamaño n+1 con n la cantidad de filas de la matriz) se almacena la posición de la primera ocurrencia de cada fila

- c) Es un conjunto de reglas que deben cumplir un emisor y un receptor para comunicarse a través de una red.
- d) signo: 0 positivo  
 exponente:  $10000001 = 129 \Rightarrow \text{exp} = 129 - 127 = 2$   
 mantisa: 1,1101  
 $\Rightarrow (\text{decimal}) 111,01 = 4 + 2 + 1 + 0.25 = 7.25$
- e) 13.5 en binario es:  $1101.1 \Rightarrow 1.1011 \times 2^3$   
 signo=1  
 $e=3 + 127 = 130 \rightarrow$  binario 10000010  
 $f=101100000000000000000000$   
 1 10000010 101100000000000000000000

|                   |                   |  |
|-------------------|-------------------|--|
| <b>Problema 2</b> | 15 (6, 6, 3) ptos |  |
|-------------------|-------------------|--|

```

a) function [Bf,Bc,Bv] = ej2p1 (Af,Ac,Av,x)
    n = length(Af);

    Bf = [];
    Bc = [];
    Bv = [];

    for i=1:n
        if ((Ac(i) >= Af(i)) & (Av(i) > x))
            Bf = [Bf, Af(i)];
            Bc = [Bc, Ac(i)];
            Bv = [Bv, Av(i)];
        end
    end
end

```

```

b) function [Bf,Bc,Bv] = ej2p2 (Af,Ac,Av,x)
    n = length(Af);

    if n == 0
        Bf = [];
        Bc = [];
        Bv = [];
    else
        [Bf,Bc,Bv] = ej2b2(Af(2:n),Ac(2:n),Av(2:n),x);
        if ((Ac(1) >= Af(1)) & (Av(1) > x))
            Bf = [Af(1), Bf];
            Bc = [Ac(1), Bc];
            Bv = [Av(1), Bv];
        end
    end
end

c) function ts = ej2p3 (Af,Ac,Av)
    n = length(Af);

    ts = 1;
    i = 1;

    while ((i <= n) & (ts == 1))
        if ( Ac(i) >= Af(i) )
            i = i + 1;
        else
            ts = 0;
        end
    end
end

```

|                   |                     |  |
|-------------------|---------------------|--|
| <b>Problema 3</b> | 19 (6, 5, 4, 4) pts |  |
|-------------------|---------------------|--|

```

a) function pos = ultOcurr(v,x)
    n = length(v);
    if n == 0
        pos = -1;
    elseif v(n) == x
        pos = n;
    else
        pos = ultOcurr(v(1:n-1),x);
    end

b) function pos = ultOcurrIter(v,x)
    n = length(v);
    pos = -1;
    i = n;
    while i > 0 & pos == -1
        if v(i) == x
            pos = i;
        else
            i = i-1;
        end
    end
end

```

```
c) function w = borrarPrim(v,x)
    n = length(v);
    if n == 0
        w = [];
    elseif v(1) == x
        w = v(2:n);
    else
        w = [v(1) borrarPrim(v(2:n),x)];
    end

d) function w = borrarTodos(v,x)
    n = length(v);
    w = [];
    for i=1:n
        if v(i) ~= x
            w = [w v(i)];
        end
    end
end
```

|                   |                  |  |
|-------------------|------------------|--|
| <b>Problema 4</b> | 14 (3, 4, 7) pts |  |
|-------------------|------------------|--|

a) function fi=proporcionAurea(n)

```
    fi=1
    for i=2:n
        fi=1 +1/fi;
    end
```

b) function fi= proporcionAurea(n)

```
    if n==1
        fi=1;
    else
        fi= 1 + (1/proporcionAurea(n-1))
    end
```

c) Alternativa 1

```
function paso=enQuePaso(tol)
    paso=cabecal(tol,1,1);

function paso=Cabecal(tol,pasoActual,fiAnterior)
    fi=(1 + sqrt(5))/2;
    if abs(fiAnterior-fi)<tol
        paso=pasoActual;
    else
        fiActual=1 + 1/fiAnterior;
        paso=cabecal(tol,pasoActual+1,fiActual)
    end
```

Alternativa 2

```
function paso=enQuePaso(tol)
    paso=cabecal(tol,1);

function paso=cabecal(tol,fiAnterior)
    fi=(1 + sqrt(5))/2;
    if abs(fiAnterior-fi)<tol
        paso = 1;
    else
        paso = 1 + cabecal(tol, 1 + 1/fiAnterior);
    end
```

