EJERCICIO 1 (4 puntos) (Se recomienda comenzar leyendo la nota al final del enunciado)

Consideramos el vector N con m componentes y el vector R que contiene m valores en el conjunto {0,1}, que se han obtenido como el resto de dividir las componentes de los elementos del vector N entre 2. Tomando como datos de entrada: el vector N, el vector R, una variable m (dimensión de ambos vectores), se desea realizar un algoritmo que realice los siguientes pasos:

• Construir dos vectores llamados: Par, Impar, que contengan respectivamente los elementos pares e impares del vector N.

• Obtener los valores P, Q que sean, respectivamente, los valores máximos almacenados en los vectores Par e Impar.

• Construir dos matrices, llamadas NP (de m filas y P columnas) y NI (de m filas y Q columnas), que contengan, respectivamente, números naturales pares consecutivos comenzando en el número 2 y números naturales impares consecutivos comenzando en el número 1.

NOTA: Al dividir un número entre 2 el resto es 0 si es par, y 1 si es impar.

Ejemplo:

$$\mathbf{N} = (8 \quad 5 \quad 6 \quad 7); \text{Par} = (8 \quad 6); \text{Im par} = (5 \quad 7);\ P = 8;\ Q = 7$$

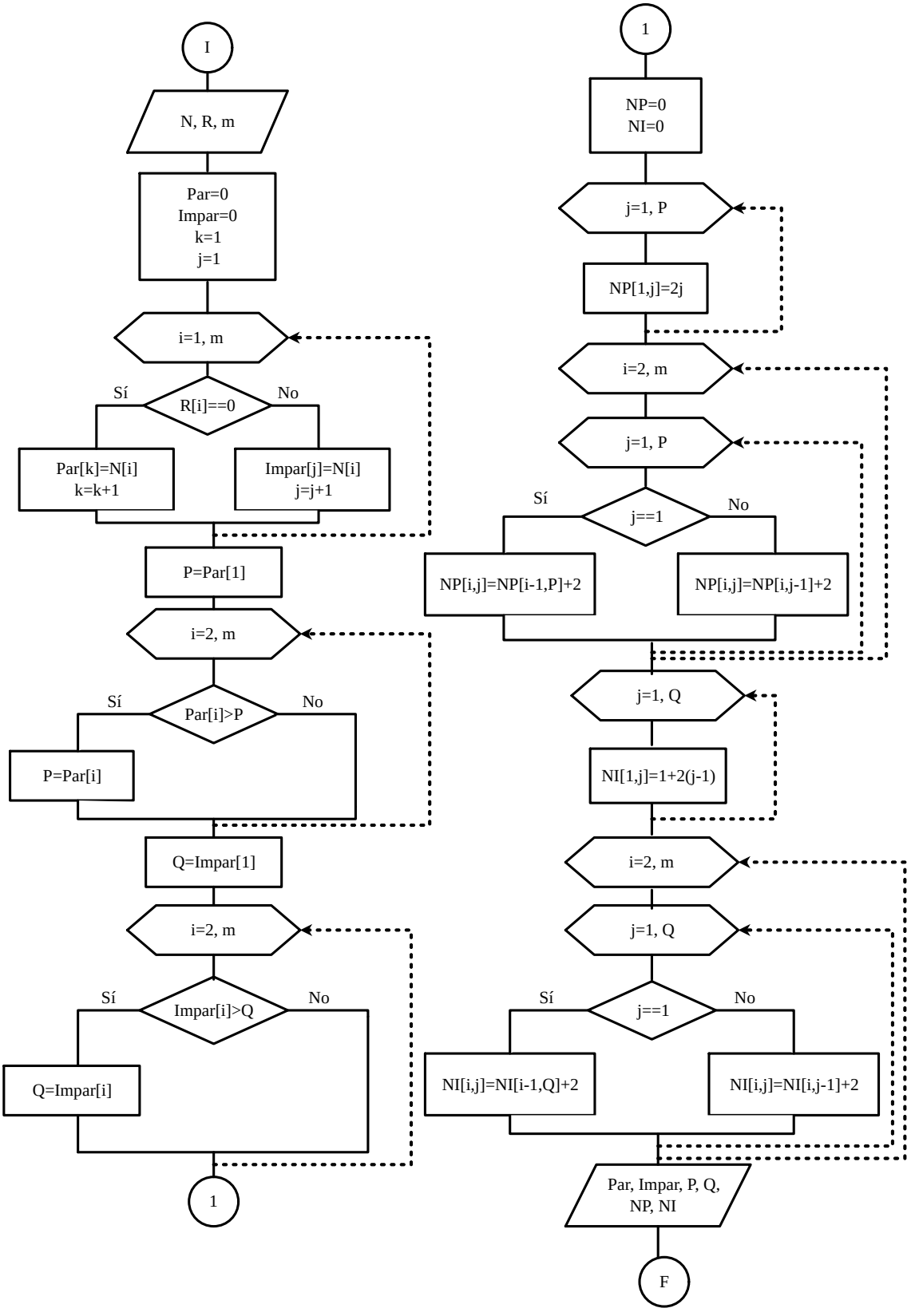
$$\mathbf{NP} = \begin{pmatrix} 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \\ 18 & 20 & 22 & 24 & 26 & 28 & 30 & 32 \\ 34 & 36 & 38 & 40 & 42 & 44 & 46 & 48 \\ 50 & 52 & 54 & 56 & 58 & 60 & 62 & 64 \end{pmatrix}; \quad \mathbf{NI} = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 29 & 31 & 33 & 35 & 37 & 39 & 41 \\ 43 & 45 & 47 & 49 & 51 & 53 & 55 \end{pmatrix}$$

Este es un ejercicio considerablemente largo en el que tenemos que hacer uso de bucles for, estructuras condicionales y los ejercicios básicos de algoritmia.

En el primer apartado, tenemos que crear un bucle desde i hasta m para que recorra todos los elementos que contiene el vector R, donde se han almacenado los restos de la división de los elementos de N entre 2. Si R[i] es 0, implica que N[i] es un número par, y por tanto lo tendremos que almacenar en el vector Par. Si por el contrario, R[i] no es 0, implica que N[i] es impar y lo tendremos que almacenar en el vector impar. Es muy importante que no queden ceros en los vectores Par e Impar, y por tanto hay que crear dos subíndices, k para el vector Par y j para el Impar. Dichos subíndices se inicializarán a 1 y, cuando se cumpla la condición, después de hacer Par[k]=N[i] (o bien, Impar[j]=N[i]), tendremos que sumarle 1 a dicho subíndice, para que el siguiente valor de N que cumpla la condición se almacene en la siguiente posición del correspondiente vector.

El segundo apartado es uno de los ejercicios más básicos de algoritmia: tenemos que asignar a P el valor de Par[1], y después, mediante un bucle, comparar cada valor de Par[i] con P. Si es mayor, se guardará en P el valor de Par[i], y así sucesivamente. Es igual para Q.

En el último, para la matriz NP hemos decidido hacer por separado la primera fila, en la que cada elemento se obtiene multiplicando por 2 la columna en la que está. Para el resto de filas, si es el primer elemento de la misma, lo obtenemos, y si no, como la suma de 2 y el valor de la misma fila y columna anterior. Para NI hacemos igualmente la primera fila por separado, como 1+ 2(columna-1), que como podemos comprobar se cumple para toda la fila. Al igual que para NP, para las demás filas, el primer elemento de la fila se obtiene sumando 2 al valor de la fila anterior, última columna; y el resto como 2+el elemento de la misma fila y columna anterior.


I
N, R, m
Par=0
Impar=0
k=1
j=1
i=1, m
Sí
R[i]==0
No
Par[k]=N[i]
k=k+1
Impar[j]=N[i]
j=j+1
P=Par[1]
i=2, m
Sí
Par[i]>P
No
P=Par[i]
Q=Impar[1]
i=2, m
Sí
Impar[i]>Q
No
Q=Impar[i]
1
1
NP=0
NI=0
j=1, P
NP[1,j]=2j
i=2, m
j=1, P
Sí
j==1
No
NP[i,j]=NP[i-1,P]+2
NP[i,j]=NP[i,j-1]+2
j=1, Q
NI[1,j]=1+2(j-1)
i=2, m
j=1, Q
Sí
j==1
No
NI[i,j]=NI[i-1,Q]+2
NI[i,j]=NI[i,j-1]+2
Par, Impar, P, Q,
NP, NI
F

EJERCICIO 2 (4 puntos) Se desea realizar un algoritmo tal que, dadas tres variables M, n, N permita obtener los elementos de cierta matriz C de N filas y M columnas cuyos elementos vienen dados por la expresión: $C_{i,j} = \frac{j^2}{2^n} \sum_{k=0}^{n} (comb * (x_i + 1)^{n-k} (x_i - 1)^k)$, (j=1,…, M; i=1, ..., N)

siendo comb el número combinatorio de n sobre k cuya expresión viene dada por:

$$comb = \frac{n!}{(n-k)!k!}$$

El cálculo de los factoriales que aparecen en esta expresión debe ser incorporado al algoritmo, teniendo en cuenta que:

• Para el cálculo de k! : SI k ≤1 asignaremos k!=1.

• Para el cálculo de (n-k)!: SI k=n asignaremos (n-k)! = 1

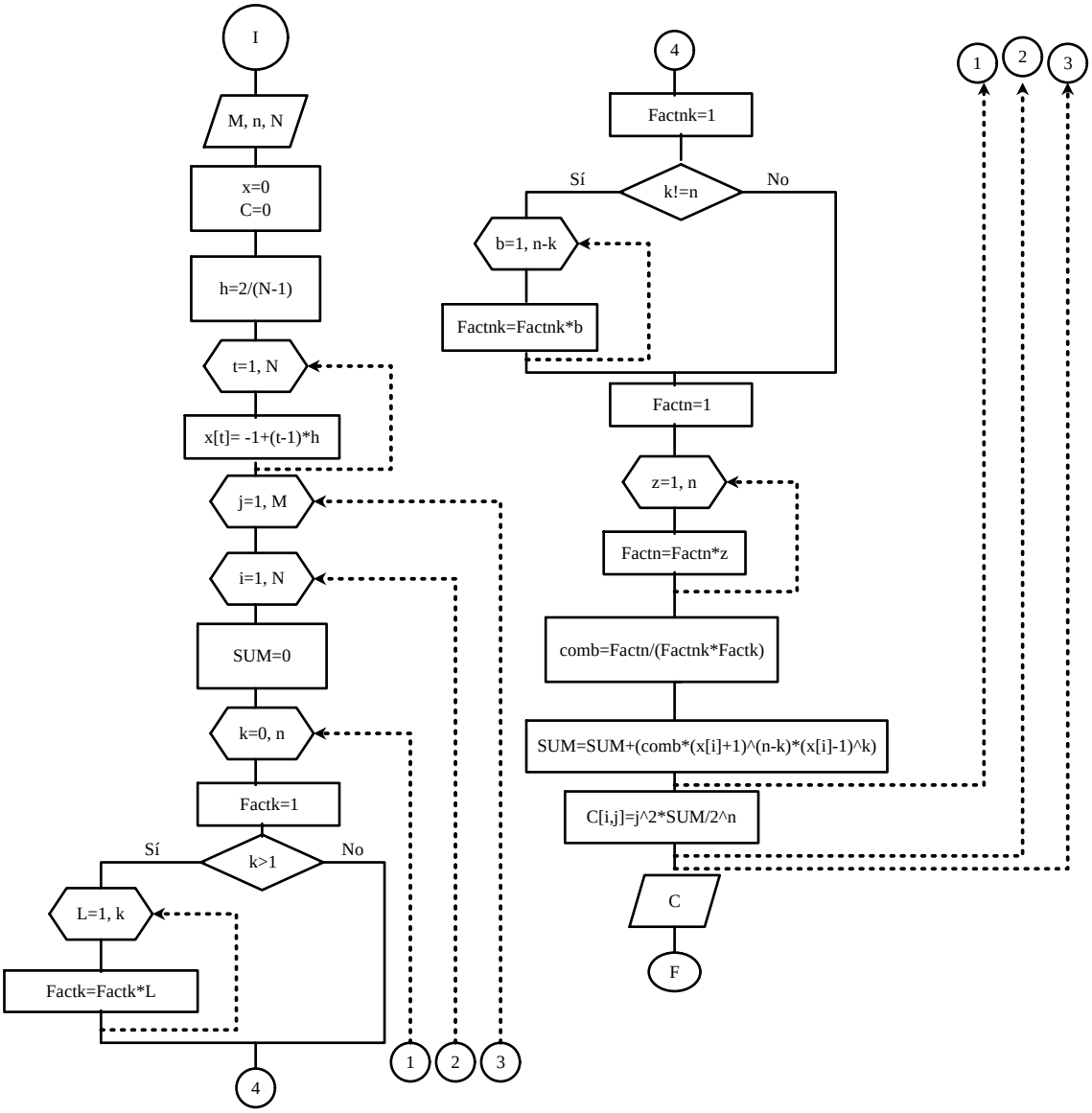
Las N componentes del vector x se obtendrán uniformemente distribuidas en el intervalo [-1,1].

Para este ejercicio empezamos calculando el vector x, cuyos elementos están uniformemente distribuidos. Tendremos que obtener h, haciendo la operación 2/(N-1), pues N es el número de elementos y N-1 el número de intervalos que definen, mientras que 2 es el resultado de restar los extremos, 1 y -1. Posteriormente tenemos que hacer un bucle que asigne a x[t] el resultado de -1 (el extremo de la izquierda) + h(t-1), es decir, el extremo de la izquierda más la longitud de todos los intervalos hasta el elemento considerado.

Posteriormente tenemos que crear los bucles anidados i y j que dice el enunciado, primero el de j (que es el primer subíndice que aparece) y después el de i, posteriormente crearemos el de k. Dentro de ellos pasamos al cálculo de factoriales, Factk, Factnk y Factn, que como productorios que son se inicializan a 1. Factn es inmediato, hacemos un bucle que con z desde 1 hasta n; Factn se obtendrá como Factn*z. Después tenemos Factk, que es también un productorio, pero tenemos que controlar que si k<=1, su factorial será 1. Por tanto, primero creamos una estructura condicional k>1, y si se cumple, tendremos el bucle con L desde 1 hasta k, y Factk será igual a Factk*L. Si no se cumple, el algoritmo no hará nada ya que hemos asignado previamente que Factk es 1.

Para Factnk tendremos que hacer algo muy parecido, aunque en este caso la condición será distinta y el bucle irá hasta n-k.

Finalmente, una vez calculados los factoriales, hacemos el sumatorio, inicializado antes a 0, mediante una variable llamada SUM. Una vez cerrado el bucle de k, simplemente terminaremos de calcular el valor de C[i,j] con la fórmula dada en el enunciado, sustituyendo todo el sumatorio por la variable SUM.


I
M, n, N
x=0
C=0
h=2/(N-1)
t=1, N
x[t]= -1+(t-1)*h
j=1, M
i=1, N
SUM=0
k=0, n
Factk=1
Sí
k>1
No
L=1, k
Factk=Factk*L
4
1
2
3
4
Factnk=1
Sí
k!=n
No
b=1, n-k
Factnk=Factnk*b
Factn=1
z=1, n
Factn=Factn*z
comb=Factn/(Factnk*Factk)
SUM=SUM+(comb*(x[i]+1)^(n-k)*(x[i]-1)^k)
C[i,j]=j^2*SUM/2^n
C
F
1
2
3