

Recurso 1: Explicación de cómo asegurarse que un vector tenga siempre el número de “espacios” necesarios cuando este depende de dos variables a partir de un ejercicio estudiado en clase.

Grupo T1

Para este primer recurso hemos decidido explicar uno de los ejercicios propuestos por el profesor, ya que uno de los integrantes del grupo no lo llegó a entender y, adicionalmente, creemos que será útil para personas que vayan a tener que estudiar esta asignatura en los próximos años o incluso para las personas de nuestra clase que no lo llegaron a entender.

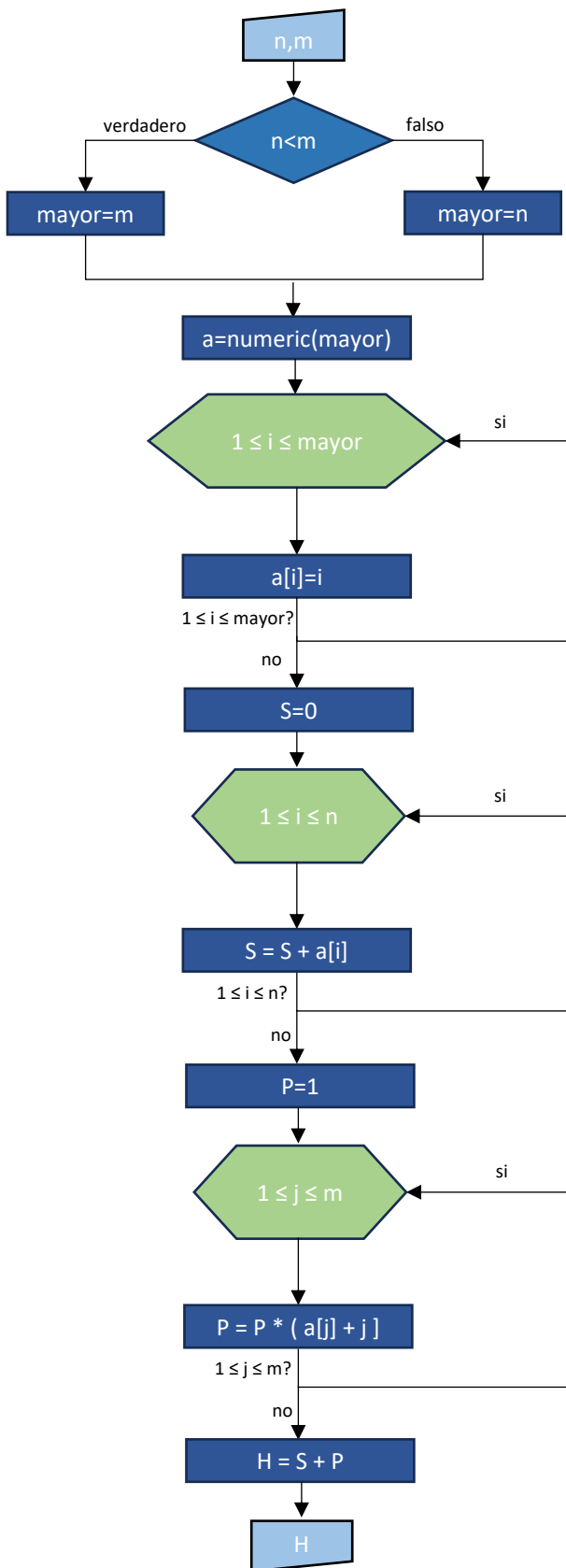
El ejercicio a explicar es el siguiente: $H = \sum_{i=1}^n A[i] + \sum_{j=1}^m (A[j] + j)$, y nos hemos centrado sobre todo en la parte de los límites del sumatorio, ya que era la parte que creemos que tenía más dificultad del problema, y la que la integrante del grupo no llegó a comprender del todo.

Para explicarlo, hemos realizado en primer lugar un diagrama de flujo, ya que es más visual y fácil de entender que si tenemos directamente la ecuación programada en R. Tanto el diagrama de flujo como el programa en R vienen anotados con explicaciones que facilitan aún más el entendimiento.

Control de calidad

La integrante del grupo que no había conseguido entender el problema en clase acabó comprendiéndolo con el recurso realizado.

$$H = \sum_{i=1}^n A[i] + \sum_{j=1}^m (A[j] + j)$$

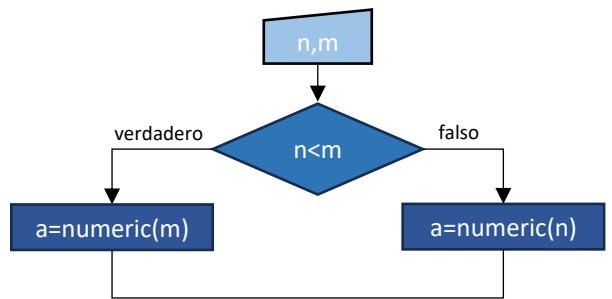


Como vamos a definir un único vector ("a") que va a depender de dos variables distintas ("n" y "m"), es necesario asegurarnos que el vector va a tener todos los "espacios" posibles. Para ello, después de introducir los valores de "n" y "m", vamos a crear una nueva variable* ("mayor") con la que definiremos el vector, la cual tendrá el mismo valor que "n" o "m" (dependiendo de cuál sea más grande).

Con este bucle "for" conseguimos crear el vector "a" con el número máximo de espacios que necesitaremos en el futuro (ya sea el de n o el de m)

Al crear el sumatorio ("S"), y más adelante el productorio ("P"), ya podemos poner el número de "espacios" reales q necesita rellenar; por lo que, en sus respectivos bucles "for" utilizaremos las variables "n" y "m"

* si no creásemos una nueva variable y lo pusiesemos de la siguiente forma:



Esto nos daría lugar a un error, ya que al continuar con el programa no nos estaríamos asegurando de que el vector tiene el número de espacios necesarios para "S" y "P". Por ejemplo, si se da el caso de que $n < m$; en el bucle "for" para el sumatorio en el que pongamos $1 \leq i \leq n$, entonces el vector ya no tendría los espacios necesarios para el productorio.

Programa en R

```
#limpiar consola
cat("\014")
#limpiar datos en memoria principal
rm(list=ls())

#preguntar n (n° de vueltas q da el sumante) & m (n° de vueltas q da el
factorial)
n <- as.numeric(readline(prompt = "Introduce el valor de n: "))
m <- as.numeric(readline(prompt = "Introduce el valor de m: "))

#definir si m o n es más grande
#si se hace en un bucle if & se introduce otro valor, no se pierde el
valor menor

if (n<m) {

    mayor = m

} else {

    mayor = n

}

#definir vector con el valor más grande
#para q sea capaz de usar todos los "slots" si es necesario
#aunque en le bucle para el n° menor no se usen todos los slots xq el
bucle para

a <- numeric(mayor)

#tmb se puede definir el vector de forma definida

# tmb puedo definir el vector manualmente con valores específicos

# a <- c(2, 4, 6, 8, 10, 12)

# Verificar si el número de elementos en el vector es suficiente

# if (length(a) < mayor) {

    #stop("Error: El valor de 'mayor' es mayor que la cantidad de elementos en
el vector.")

#}

# tmb puedo definir el vector usando la función seq() que genera una
secuencia

# a <- seq(1, mayor)

# Esto es equivalente a llenar el vector con números del 1 al valor de
'mayor'

for (i in 1:mayor) {

    a[i]=i

}

#definir sumante
S=0

#bucle for hasta n, xq el sumante va hasta n
```

```
#aunq el vector sea capaz de utilizar los n° de la variable mayor
#se pide que utilice solo los de la variable n, ya sea la mayor o menor
for (i in 1:n) {

    S = S + a[i]

}

#definir factorial
P=1

#bucle for hasta m, xq el factorial va hasta m
#aunq el vector sea capaz de utilizar los n° de la variable mayor
#se pide que utilice solo los de la variable m, ya sea la mayor o menor

for (j in 1:m) {

    P = P * (a[j]+ j)

}

#sumar sumante y factorial
H=S+P

#imprimir resultado
print(H)
```