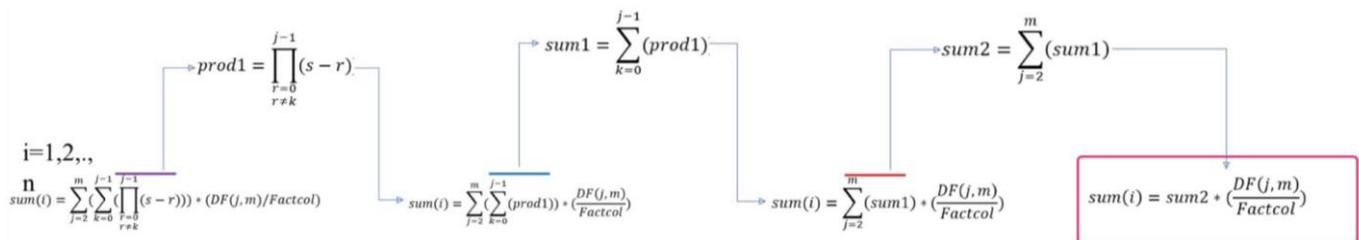


EJEMPLO CAMBIO DE ALGORITMO A R

Este algoritmo proviene de la siguiente fórmula que representa una operación de sumatorios y productorios:

$$i=1,2,.., \quad \sum_{j=2}^n \left(\sum_{k=0}^m \left(\prod_{\substack{r=0 \\ r \neq k}}^{j-1} (s-r) \right) \right) * (DF(j,m)/Factcol)$$

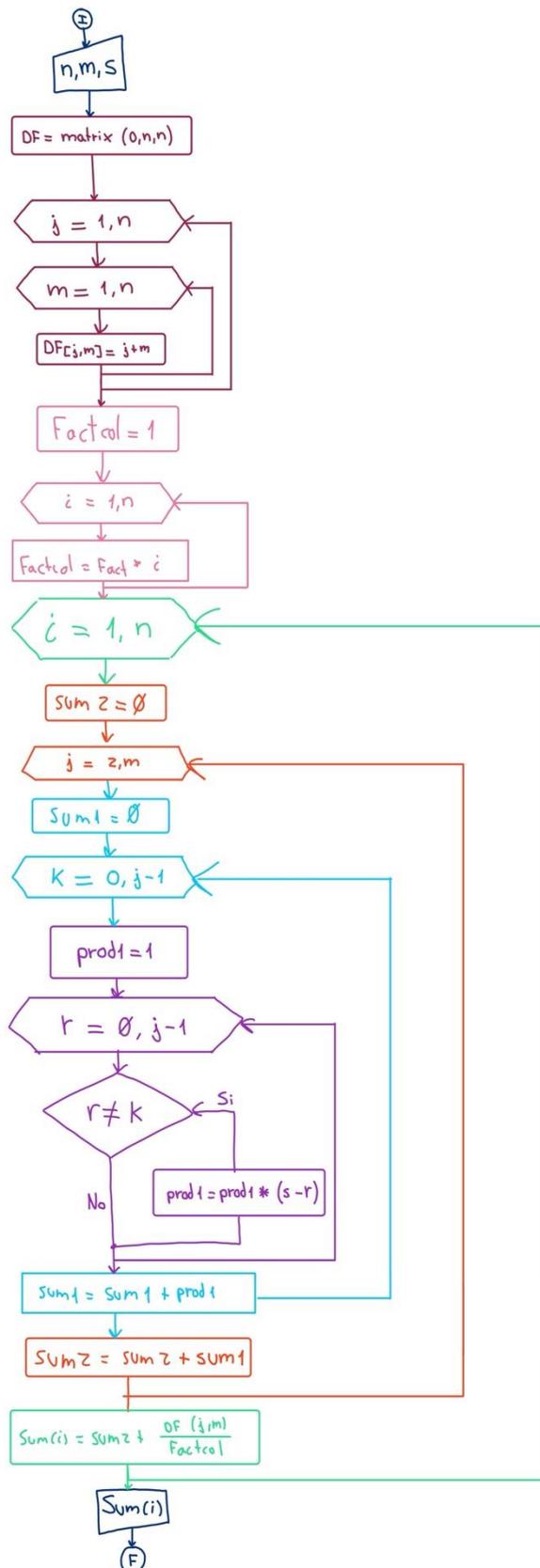
Que simplificada sería:



Al representar esta fórmula como un algoritmo tenemos que introducir las variables de entrada (n,m,s). A continuación, tenemos que hacer dos bucles for para obtener la matriz “DF”, y un bucle for para obtener el factorial “Factcol”. Una vez hemos obtenido esto, tenemos que realizar las operaciones generales. Para ello utilizaremos una concatenación de bucles for que representa el “sumatorio i” que es un sumatorio del “sum2” (suma de “sum1” por el “prod”) El productorio representa la multiplicación por la matriz entre el factorial.

El algoritmo es el siguiente:

Es una versión adaptada del de clase, incluyendo el algoritmo de la matriz en vez de ponerla como variable de entrada. En clase no especificábamos para simplificar dando por hecho que el que mete los datos sabe qué es DF, pero al hacerlo en R es necesario especificar y escribir el código para la matriz y su función.



El código en R es el siguiente:

```

n<-as.numeric(readline(prompt = "Introduce el valor de n: "))
m<-as.numeric(readline(prompt = "Introduce el valor de m: "))
s<-as.numeric(readline(prompt = "Introduce el valor de s: "))

# Introducción de la matriz DF
DF<-matrix(0,nrow = n,ncol = n)

for (j in 1:n){
  for (m in 1:n){
    DF[j,m]=j+m
  }
}

factcol=1
for (i in 1:n){
  factcol=factcol*i
}

sum <- numeric(n)
for (i in 1:n){
  sum2=0
  for (j in 2:m){
    sum1=0
    for (k in 0:(j-1)){
      prod1=1
      for (r in 0:(j-1)){
        while (r!= k) {
          prod=prod1*(s-k)
        }
      }
      sum1=sum1+prod
    }
  }
}

```

```

    }
    sum2=sum2+sum1
  }
  sum[i]=sum2*(DF[j,m])/factcol
}
resultado<-sum(n)
print(resultado)

```

Explicación:

1º - Introducimos las variables de entrada como “as.numeric” para que R las pueda leer y utilizamos “(readline(prompt = "Introduce el valor de : "))” para que le pida al usuario que introduzca los datos.

2º- Vamos a calcular la matriz “DF”. Inicializamos la matriz “DF” a 0 y como depende de las variables j y m hacemos un bucle for por cada una y añadimos la fórmula $j+m$.

3º - Vamos a calcular el factorial “Factcol”. Al ser un productorio lo inicializamos a 1, después hacemos un bucle for.

Ya tendríamos las variables y calculada la matriz y el factorial, ahora podemos empezar con los sumatorios.

4º- Empezamos con el sumatorio principal, el resultado de la fórmula, “sum (i)”. Como solo tiene un subíndice solo es necesario hacer un bucle for de 1 a n. Después al final añadimos la fórmula igualando el sumatorio a la fórmula que habíamos simplificado.

5º- Inicializamos el segundo sumatorio “sum2” a 0 y hacemos un bucle for que vaya de 2 a m. Después al final añadimos la fórmula que corresponde a la suma de este sumatorio más el “sum1”.

6º- Inicializamos el siguiente sumatorio “sum1” a 0 y hacemos un bucle for que vaya de 0 a j-1. Después al final añadimos la fórmula, igualando “sum1” a la suma de este sumatorio y el productorio.

7º- Inicializamos el productorio “prod1” a 1, al contrario que los sumatorios, a continuación, introducimos un bucle for que vaya de 0 a j-1. El enunciado nos plantea una condición especial en el productorio, r debe ser distinta a k. Para

escribir esta condición en R introducimos un bucle while, en el que si r y k son distintas te lleve a la fórmula “ $prod=prod1*(s-k)$ ”, pero si son iguales continúe con el recorrido.

8º- Finalmente tras continuar con el recorrido realizando todas las fórmulas indicadas en cada sumatorio obtenemos el resultado final que sería $sum(n)$, y le pedimos que nos lo muestre con `print(resultado)`