
Alumno: **PAREDES AGUILERA CHRISTIAN LIMBERT.**
C.I.: **6788578 L.P.**
Universidad: **Mayor de San Andrés.**
Carrera: **Matemáticas.**
Asignatura: **Computación Científica II.**
Tarea: **9 / semana 2**
Fecha **10-11-2021**

1. COEFICIENTE BINOMIAL (Capítulo 2, problema 3. Michael Spivak, Calculo Infinitesimal.)

a) Si $0 \leq k \leq n$, se define el coeficiente binomial $\binom{n}{k}$ por

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{k!}, \quad \text{si } k \neq 0, n$$

b) Código fuente.

función de coheficiente binomial

```
def binomial(n, k):  
    if k < 0 or k > n:  
        return 0  
    if k == 0 or k == n:  
        return 1  
    t = 1  
    for i in range(min(k, n - k)):  
        t = t * (n - i) // (i + 1)  
    return t
```

c) Prueba de la ejecución del programa.

```
rode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,20))'  
231  
rode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,0))'  
1  
rode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(22,4))'  
7315  
rode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(2233,34))'  
1915992310120937079161553621708885020654666583096334247860059756882159911700  
rode@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'import coe_binom; print(coe_binom.binomial(6,5))'  
6
```

2. NÚMERO NATURAL PAR O IMPAR (Capítulo 2, problema 8. Michael Spivak, Calculo Infinitesimal.)

a) Demostrar que todo número natural es o par o impar.

Demostración.- Asumimos que n es impar o par, entonces debemos probar que $n + 1$ también, es o bien impar o bien par.

Sea n par entonces $n = 2k$ para algún k . Así $n + 1 = 2k + 1$ y por definición vemos que es impar.

Luego sea n impar, entonces $n = 2k + 1$ para algún k . Por lo tanto $n + 1 = 2k + 1 + 1 = 2k + 2 = 2(k + 1)$.

Así en cualquiera de los dos casos, $n + 1$ es o bien par o impar.

b) Código fuente.

```
# Todo número natural es par o impar

def parImpar(k):
    par, impar = [], []
    for i in range(1, k+1):
        if i % 2 == 0:
            par.append(i)
        else:
            impar.append(i)
    return "Par: {} \nImpar: {}".format(par, impar)
```

c) Prueba de la ejecución del programa.

```
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from par_impar import parImpar; print(parImpar(5))'
Par: [2, 4]
Impar: [1, 3, 5]
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from par_impar import parImpar; print(parImpar(11))'
Par: [2, 4, 6, 8, 10]
Impar: [1, 3, 5, 7, 9, 11]
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from par_impar import parImpar; print(parImpar(50))'
Par: [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]
Impar: [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]
fede@ubuntu:/d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from par_impar import parImpar; print(parImpar(65))'
Par: [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]
Impar: [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, 57, 59, 61, 63, 65]
```

3. DESIGUALDAD DE BERNOULLI (Capítulo 2, problema 19. Michael Spivak, Calculo Infinitesimal.)

a) Demostrar la desigualdad de Bernoulli: Si $h > -1$, entonces

$$(1+h)^n \geq 1+nh$$

¿Por qué es esto trivial si $h > 0$?

Demostración.- Si $n = 1$, entonces $(1+h)^n = 1+nh$. Supóngase que $(1+h)^n \geq 1+nh$. Entonces $(1+h)^{n+1} = (1+h)(1+h)^n \geq (1+h)(1+nh)$, ya que $1+h > 0$ luego $1+(n+1)h+nh^2 \geq 1+(n+1)h$. Para $h > 0$, la igualdad se sigue directamente del teorema del binomio, ya que todos los demás términos que aparecen en el desarrollo de $(1+h)^n$ son positivos.

b) Código fuente.

```
# Desigualdad de Bernoulli
def Bernoulli(n,h):
    if h>-1:
        return "{} >= {}".format((1+h)**n,1+n*h)
    else:
        return "h debe ser mayor que -1"
```

c) Prueba de la ejecución del programa.

```
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(74,3))'
356811923176489970264571492362373784095686656 >= 223
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(44,30))'
416787349596085853594397793593920583748646209303727133530079150721 >= 1321
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(4,3))'
256 >= 13
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(3,3))'
64 >= 10
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(3,34))'
42875 >= 103
fede@ubuntu: /d/git/matematicas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from bernoulli import Bernoulli; print(Bernoulli(32,42))'
18663392025969146670022260524972832947786731075670401 >= 1345
```

4. SUCESIÓN DE FIBONACCI (Capítulo 2, problema 20. Michael Spivak, Calculo Infinitesimal.)

a) La sucesión de Fibonacci a_1, a_2, a_3, \dots se define como sigue:

$$\begin{aligned}a_1 &= 1, \\a_2 &= 1, \\a_n &= a_{n-1} + a_{n-2} \quad \text{para } n \geq 3\end{aligned}$$

Esta sucesión, cuyos primeros términos son 1, 1, 2, 3, 5, 8, ..., fue descubierta por Fibonacci (1175-1250, aprox.) en relación con un problema de conejos. Fibonacci supuso que una pareja de conejos criaba una nueva pareja cada mes y que después de dos meses cada nueva pareja se comportaba del mismo modo. El número a_n de parejas nacidas en el n -ésimo mes es $a_{n-1} + a_{n-2}$, puesto que nace una pareja por cada pareja nacida en el mes anterior, y además, cada pareja nacida hace dos meses produce ahora una nueva pareja. Es verdaderamente asombroso el número de resultados interesantes relacionados con esta sucesión, hasta el punto de existir una Asociación de Fibonacci que publica una revista, the Fibonacci Quarterly.

Demostrar que

$$a_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

Demostración.- Al ser

$$\frac{\left(\frac{1+\sqrt{5}}{2}\right)^1 - \left(\frac{1-\sqrt{5}}{2}\right)^1}{\sqrt{5}} = \frac{\sqrt{5}}{\sqrt{5}} = 1$$

La fórmula es válida para $n = 1$ y también se cumple para $n = 1$. Supóngase que es válida para todo $k < n$, donde $n \geq 3$. En tal caso es válida en particular para $n - 1$ y para $n - 2$, luego por hipótesis

$$\begin{aligned}a_n &= a_{n-1} + a_{n-2} \\&= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}}{\sqrt{5}} + \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}}{\sqrt{5}} \\&= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} \left(\frac{1+\sqrt{5}}{2}\right) - \left(1 - \frac{1-\sqrt{5}}{2}\right)^{n-2} \left(1 - \frac{1-\sqrt{5}}{2}\right)}{\sqrt{5}} \\&= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} \left(\frac{1+\sqrt{5}}{2}\right)^2 - \left(1 - \frac{1-\sqrt{5}}{2}\right)^{n-2} \left(1 - \frac{1-\sqrt{5}}{2}\right)^2}{\sqrt{5}} \\&= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}\end{aligned}$$

b) Código fuente.

```
# fibonacci

def fib(n):
    a, b = 0, 1
    for _ in range(n):
        a, b = b, a+b
    sucesion = (((1+5**(1/2))/2)**n-((1-5**(1/2))/2)**n)/5**(1/2)
    return "Fibonacci: {} \nSucesión de Fibonacci: {}".format(a,sucesion)
```

c) Prueba de la ejecución del programa.

```
rode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from fibo import fib; print(fib(70))'
Fibonacci: 190392490709135
Sucesión de Fibonacci: 190392490709135.44
rode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from fibo import fib; print(fib(5))'
Fibonacci: 5
Sucesión de Fibonacci: 5.000000000000001
rode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from fibo import fib; print(fib(21))'
Fibonacci: 10946
Sucesión de Fibonacci: 10946.000000000007
rode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from fibo import fib; print(fib(10))'
Fibonacci: 55
Sucesión de Fibonacci: 55.00000000000014
```

5. DESIGUALDAD DE SCHWARZ (Capítulo 2, problema 21. Michael Spivak, Calculo Infinitesimal.)

a) La desigualdad de Schwarz (problema 1-19) tiene en realidad una forma más general:

$$\sum_{i=1}^n x_i y_i \leq \sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}$$

Dar de esto tres demostraciones, análogas a las tres demostraciones del problema 1-19

Demostración.-

I) Como antes, la demostración es trivial si para todo $y_i = 0$ o si hay algún número λ con $x_i = \lambda y_i$ para todo i . Es decir,

$$0 < \sum_{i=1}^n n(\lambda y_i - x_i)^2$$

$$= \lambda \left(\sum_{i=1}^n y_i^2 \right) - 2\lambda \left(\sum_{i=1}^n x_i y_i \right) + \sum_{i=1}^n x_i^2 \quad \text{así por el problema 1-18 queda demostrado.}$$

II) Usando $2xy \leq x^2 + y^2$ con

$$x = \frac{x_i}{\sqrt{\sum_{i=1}^n x_i^2}}, \quad y = \frac{y_i}{\sqrt{\sum_{i=1}^n y_i^2}}$$

obtenemos,

$$\frac{2x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \leq \frac{x_i^2}{\sum_{i=1}^n x_i^2} + \frac{y_i^2}{\sum_{i=1}^n y_i^2} \quad (1)$$

luego

$$\frac{\sum_{i=1}^n 2x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \leq \frac{\sum_{i=1}^n x_i^2}{\sum_{i=1}^n x_i^2} + \frac{\sum_{i=1}^n y_i^2}{\sum_{i=1}^n y_i^2} = 2$$

Nuevamente, la igualdad se cumple solo si se cumple en (1) para todo i , lo que significa que,

$$\frac{x_i}{\sqrt{\sum_{i=1}^n x_i^2}} = \frac{y_i}{\sqrt{\sum_{i=1}^n y_i^2}}$$

para todo i . Si todo y_i es distinto de 0. Esto significa que $x_i = \lambda y_i$ para

$$\lambda = \frac{\sqrt{\sum_{i=1}^n x_i^2}}{\sqrt{\sum_{i=1}^n y_i^2}}$$

III) La demostración depende de la siguiente igualdad:

$$\sum_{i=1}^n x_i^2 \cdot \sum_{i=1}^n y_i^2 = \left(\sum_{i=1}^n x_i y_i \right)^2 + \sum_{i < j} (x_i y_j - x_j y_i)^2$$

al verificar esta igualdad notamos que,

$$\sum_{i=1}^n x_i^2 \cdot \sum_{i=1}^n y_i^2 = \sum_{i=1}^n x_i^2 y_i^2 + \sum_{i \neq j} x_i^2 y_j^2$$

y por lo tanto,

$$\left(\sum_{i=1}^n x_i y_i \right)^2 = \sum_{i=1}^n (x_i y_i)^2 + \sum_{i \neq j} x_i y_i x_j y_j$$

La diferencia es

$$\begin{aligned} \sum_{i \neq j} (x_i^2 y_j^2 - x_i y_i x_j y_j) &= 2 \sum_{i < j} (2_i^2 y_j^2 + x_j^2 y_i^2 - x_i y_i x_j y_j) \\ &= 2 \sum_{i < j} (x_i y_j - x_j y_i)^2 \end{aligned}$$

Si la igualdad se cumple en la desigualdad de Schwarz, para todo $x_i y_j = x_j y_i$. Si algún $y_i \neq 0$ y $y_i \neq 0$, luego $x_i = \frac{x_1}{y_1} y_i$ para todo i , así tenemos que $\lambda = \frac{x_1}{y_1}$

b) Código fuente.

```
# Desigualdad de Schwarz

def Schwarz(n):
    sum1, sum2, sum3 = 0,0,0
    for i in range(1,n+1):
        sum1+=i * i**2
        sum2+=i**2
        sum3+=(i**2)**2
    return "{} {} <= {}".format(sum1, ((sum2)**(1/2)) * ((sum3)**(1/2)))
```

c) Prueba de la ejecución del programa.

```
fode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from schwarz import Schwarz; print(Schwarz(10))'
3025 <= 3123.0121677636803
fode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from schwarz import Schwarz; print(Schwarz(100))'
25502500 <= 26338760.073426012
fode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from schwarz import Schwarz; print(Schwarz(183))'
283450896 <= 292746464.31612736
fode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from schwarz import Schwarz; print(Schwarz(1))'
1 <= 1.0
fode@ubuntu:/d/git/matenaticas/programacion/inf-111/python/tareas_mat/week2$ python3 -c 'from schwarz import Schwarz; print(Schwarz(74))'
7700625 <= 7953111.581961177
```