Una **permutación** de un conjunto de elementos es una disposición de dichos elementos **teniendo en cuenta el orden**. Una **combinación** de un conjunto de elementos es una selección de dichos elementos **sin tener en cuenta el orden.**

La diferencia entre permutaciones y combinaciones es que **en las permutaciones importa el orden de los elementos, mientras que en las combinaciones no importa el orden** en que se disponen los elementos (solo importa que estén presentes).

Permutaciones

Una permutación de un conjunto de elementos es una disposición de dichos elementos **teniendo en cuenta el orden**. El número de permutaciones de “n” elementos tomados de “k” en “k” se calcula con la fórmula:

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

Ejemplo 1:

Eduardo, Carlos y Sergio se han presentado a un concurso de pintura. El concurso otorga $200 al primer lugar y $100 al segundo. ¿De cuántas formas se pueden repartir los premios de primer y segundo lugar?

Solución:

En este caso, si importa el orden, ya que no es lo mismo quedar en primer lugar que en segundo, además, los premios son diferentes. Por ejemplo, un arreglo o disposición, es que Carlos ocupe el primer lugar y Sergio el segundo. Otro arreglo, sería que Sergio ocupe el primer lugar y Eduardo el segundo. El número total de arreglos o formas lo calculamos con la fórmula:

$$n = 3 \ (número\ total\ de\ elementos); \quad k = 2 \ (tomados\ de\ dos\ en\ dos)$$

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

$$P_2^3 = \frac{3!}{(3-2)!} = \frac{3x2x1}{1!} = \frac{6}{1} = \boldsymbol{6\ formas}$$

Combinaciones

Una combinación de un conjunto de elementos es una selección de dichos elementos **sin tener en cuenta el orden.**

El número de combinaciones de "n" elementos tomados de "k" en "k" se calcula con la fórmula:

$$C_k^n = \frac{n!}{(n-k)!\,k!}$$

Ejemplo 2:

Un chef va a preparar una ensalada de verduras con tomate, zanahoria, papa y brócoli. ¿De cuántas formas se puede preparar la ensalada usando solo 2 ingredientes?

Solución:

En este caso, no importa el orden en que se tomen los ingredientes para la ensalada, pues da igual si es una ensalada de tomate con zanahoria, que una ensalada de zanahoria con tomate, ya que al final, el chef mezclará los dos ingredientes.

Un arreglo podría ser zanahoria y tomate, otro arreglo podría ser tomate y papa, otro arreglo podría ser papa y brócoli. El problema nos indica que solo se pueden usar 2 ingredientes en la ensalada. El número total de arreglos o formas lo calculamos con la fórmula:

$$n = 4\ (\textit{número total de elementos}); \quad k = 2\ \ (\textit{tomados de dos en dos})$$

$$C_k^n = \frac{n!}{(n-k)!\,k!}$$

$$C_2^4 = \frac{4!}{(4-2)!\,2!} = \frac{4x3x2x1}{2!\,x2!} = \frac{24}{2x2} = \frac{24}{4} = 6\ formas$$

Ejemplo 3:

Se va a programar un torneo de ajedrez para los 10 integrantes de un club. ¿Cuántos partidos se deben programar si cada integrante jugará con cada uno de los demás sin partidos de revancha?

Solución:

En este torneo se van a realizar partidas de ajedrez en cada una de las cuales participan 2 jugadores. Por ello, necesitamos ordenamientos de 2 en 2, es decir, k = 2. Además, en estos ordenamientos participarán los 10 jugadores, por eso, n = 10.

En este caso, no importa el orden, ya que solo necesitamos agrupar los jugadores, es igual que juegue Jorge contra Carlos, que Carlos contra Jorge. Además, no hay partido de revancha, es una sola partida con cada contrincante.

$$C_k^n = \frac{n!}{(n-k)!\,k!}$$

$$n = 10\ (número\ de\ jugadores) \wedge k = 2\ (tomados\ de\ 2\ en\ 2)$$

$$C_2^{10} = \frac{10!}{(10-2)!\,2!} = \frac{10!}{8!\,2!} = \frac{10 \times 9 \times 8!}{8! \times 2 \times 1} = \frac{\overset{5}{\cancel{10}} \times 9 \times \cancel{8!}}{\cancel{8!} \times \underset{1}{\cancel{2}} \times 1}$$

$$C_2^{10} = 5 \times 9 = 45$$

Se deben programar 45 partidos.

Ejemplo 4:

Considera un grupo de 10 estudiantes de los cuales 4 son mujeres y 6 son hombres. De acuerdo con esa información, determine:

a) El número de formas en que se puede elegir un representante del grupo.

b) El número de formas en que se puede elegir un comité de 3 miembros, donde al menos uno de los miembros sea mujer.

Solución:

En este caso, no nos dicen que el comité tiene rangos, por lo tanto, no importa el orden. Aplicaremos la fórmula de combinaciones.

a) El número de formas en que se puede elegir un representante del grupo.

De un total de 10 miembros, elegiremos a uno:

$$C_1^{10} = \frac{10!}{(10-1)!\,1!} = \frac{10 \times \cancel{9!}}{\cancel{9!} \times 1!} = \frac{10}{1} = \mathbf{10}\ \boldsymbol{formas}$$

b) El número de formas en que se puede elegir un comité de 3 miembros, donde al menos uno de los miembros sea mujer.

Al menos uno de los 3 miembros tiene que ser mujer. Eso significa que el comité puede estar formado por 1, 2 o 3 mujeres.

– Si el comité está formato por 1 mujer, significa que de las 4 mujeres seleccionaremos una, y de los 6 hombres seleccionaremos 2.

$$C_1^4 \times C_2^6 = \frac{4!}{(4-1)!\,1!} \times \frac{6!}{(6-2)!\,2!} = \frac{4 \times \cancel{3!}}{\cancel{3!} \times 1!} \times \frac{6 \times 5 \times \cancel{4!}}{\cancel{4!} \times 2 \times 1} = 4 \times 15 = 60\ formas$$

– Si el comité está formato por 2 mujer, significa que de las 4 mujeres seleccionaremos 2, y de los 6 hombres seleccionaremos 1.

$$C_2^4 \times C_1^6 = \frac{4!}{(4-2)!\,2!} \times \frac{6!}{(6-1)!\,1!} = \frac{4 \times 3 \times \cancel{2!}}{\cancel{2!} \times 2 \times 1} \times \frac{6 \times \cancel{5!}}{\cancel{5!} \times 1} = 6 \times 6 = 36\ formas$$

– Si el comité está formato por 3 mujeres, significa que de las 4 mujeres seleccionaremos 3, y de los 6 hombres no seleccionaremos a ninguno.

$$C_3^4 \times C_0^6 = \frac{4!}{(4-3)!\,3!} \times \frac{6!}{(6-0)!\,0!} = \frac{4 \times \cancel{3!}}{1! \times \cancel{3!}} \times \frac{\cancel{6!}}{\cancel{6!} \times 1} = 4 \times 1 = 4\ formas$$

En total, tenemos:

$$60 + 36 + 4 = \mathbf{100}\ \boldsymbol{formas}$$

PERMUTACIÓN Y COMBINACIÓN EN PYTHON

Python proporciona métodos directos para encontrar permutaciones y combinaciones de una secuencia. Estos métodos están presentes en el paquete itertools.

Permutación

Primero importe la librería itertools para implementar el método de permutaciones en Python. Este método toma una lista como entrada y devuelve una lista de objetos de tuplas que contienen toda la permutación en forma de lista.

```
from itertools import permutations

perm = permutations([1, 2, 3])

for i in list(perm):
    print (i)
```

Salida:

```
(1, 2, 3)
(1, 3, 2)
(2, 1, 3)
(2, 3, 1)
(3, 1, 2)
(3, 2, 1)
```

Genera n! permutaciones si la longitud de la secuencia de entrada es n.

Si desea obtener permutaciones de longitud L, impleméntelo de esta manera.

```
from itertools import permutations

perm = permutations([1, 2, 3], 2)

for i in list(perm):
    print (i)
```

Salida:

```
(1, 2)
(1, 3)
(2, 1)
(2, 3)
(3, 1)
(3, 2)
```

Genera nCr * r! permutaciones si la longitud de la secuencia de entrada es n y el parámetro de entrada es r.

Combinación

Este método toma una lista y una entrada r como entrada y devuelve una lista de objetos de tuplas que contienen todas las combinaciones posibles de longitud r en forma de lista.

```
from itertools import combinations

comb = combinations([1, 2, 3], 2)

for i in list(comb):
    print (i)
```

Salida:

```
(1, 2)
(1, 3)
(2, 3)
```

1. Las combinaciones se generan en orden de entrada. Por tanto, si la lista de entrada está ordenada, las tuplas de combinación se producirán en orden ordenado.

```
from itertools import combinations

comb = combinations([1, 2, 3], 2)

for i in list(comb):
  print (i)
```

Salida:

```
(2, 1)
(2, 3)
(1, 3)
```

2. Los elementos se tratan como únicos en función de su posición, no de su valor. Entonces, si los elementos de entrada son únicos, no habrá valores repetidos en cada combinación.

```
from itertools import combinations

comb = combinations([2, 1, 3], 2)

for i in list(comb):
  print (i)
```

Salida:

```
(1, 1)
(1, 3)
(1, 3)
```

3. Si queremos hacer una combinación del mismo elemento con el mismo elemento, usamos combinaciones_con_reemplazo.

```
from itertools import combinations_with_replacement

comb = combinations_with_replacement([1, 2, 3], 2)

for i in list(comb):
  print (i)
```

Salida:

```
(1, 1)
(1, 2)
(1, 3)
(2, 2)
(2, 3)
(3, 3)
```

Ejemplo 1: todas las permutaciones de Python:

```
import itertools
print(list(itertools.permutations([1,2,3])))
```

Ejemplo 2: Importar módulos para combinaciones y permutaciones en Python:

```
from itertools import permutations
from itertools import combinations
```

Ejemplo 3: código de permutación y combinación en Python:

```
from itertools import combinations

comb = combinations([1, 2, 3], 2)

for i in list(comb):
    print (i)
```

Ejemplo 4: código de permutación y combinación en Python

```
from itertools import permutations

perm = permutations([1, 2, 3], 2)

for i in list(perm):
    print (i)
```