

Ley D'Hont

Datos personales

- **Apellidos:**
- **Nombre:**
- **Email:**
- **DNI:**

1. Introducción

En las elecciones, una vez ejecutado el escrutinio, se debe calcular cuántos escaños corresponden a cada lista electoral. El objetivo ideal sería repartir dichos escaños de manera proporcional a los votos obtenidos por cada lista. Como esto no es posible en general, porque rara vez las proporciones son números enteros, se persigue acercarse a dicha proporcionalidad lo más posible.

El sistema D'Hondt es un método para asignar un número de escaños dado a partir de los votos recibidos por distintas listas electorales, buscando acercar al máximo el reparto a las proporciones de votos resultantes en el escrutinio. Concretamente, este sistema va asignando escaños de uno en uno, dando cada vez el escaño a la lista cuyo cociente

$$\frac{\text{número de votos recibidos}}{\text{número de escaños recibido hasta el momento} + 1}$$

sea mayor. Sobre este problema trata este ejercicio que ahora te planteo, paso a paso.

(a) Lectura de datos

Nuestros datos de partida estarán en un archivo de texto con dos líneas: el número de escaños en la primera, y los votos recibidos por cada lista electoral, en la segunda. Para concretar al máximo, te proporciono dos ejemplos. El primero es el archivo `Datos_DHont_ej1.txt` :

```
7
340000 280000 160000 60000 15000
```

Y el segundo, `Datos_DHont_ej2.txt` :

```
21
391000 311000 184000 73000 27000 12000 2000
```

Diseña una función que carga los datos de uno de esos archivos en dos variables, una (entera) con el número de escaños que se van a repartir; otra, una lista de enteros con los votos escrutados por cada lista.

In [1]:

```
# Aquí viene la función definida por el estudiante
```

In [2]:

```
# La función anterior debe funcionar según muestran las pruebas siguientes:
```

```
escannos, votos = leer_datos("Datos_DHont_ej1.txt")  
print(escannos, votos)
```

```
escannos, votos = leer_datos("Datos_DHont_ej2.txt")  
print(escannos, votos)
```

```
7 [340000, 280000, 160000, 60000, 15000]  
21 [391000, 311000, 184000, 73000, 27000, 12000, 2000]
```

(b) Reparto

Partiendo de los datos leídos, procedemos ahora a realizar el reparto. Para ello, para cada lista electoral se calcula el cociente siguiente:

$$coc_i = \frac{V_i}{s_i + 1}$$

donde V_i es el número total de votos recibidos por la lista i , y s_i el número de escaños recibidos hasta el momento por la misma lista.

Esto se ha de realizar tantas veces como escaños se desean repartir, partiendo de 0 escaños repartidos inicialmente para cada lista, naturalmente.

Te pido una función que realice dicho reparto.

In [3]:

```
# Aquí viene la función definida por el estudiante
```

In [4]:

```
# La función anterior debe funcionar según muestran las pruebas siguientes:
```

```
escannos, votos = leer_datos("Datos_DHont_ej1.txt")  
resultado = reparto(escannos, votos)  
print(resultado)
```

```
escannos, votos = leer_datos("Datos_DHont_ej2.txt")  
resultado = reparto(escannos, votos)  
print(resultado)
```

```
[3, 3, 1, 0, 0]  
[9, 7, 4, 1, 0, 0, 0]
```

(c) Reparto, mostrando las cuentas

La nueva Ley de Transparencia obliga ahora a que los repartos muestren el proceso de reparto paso a paso. Modifica la función anterior para que muestre el proceso de reparto de cada escaño.

In [5]:



```
# Aquí viene la función definida por el estudiante
```

In [6]:



```
# La función anterior debe funcionar según muestran las pruebas siguientes:
```

```
escannos, votos = leer_datos("Datos_DHont_ej1.txt")
resultado = reparto(escannos, votos)
print(resultado, "\n")
```

```
escannos, votos = leer_datos("Datos_DHont_ej2.txt")
resultado = reparto(escannos, votos)
print(resultado, "\n")
```

```
Escanno: 0
['340000/1 = 340000.0', '280000/1 = 280000.0', '160000/1 = 160000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 1
['340000/2 = 170000.0', '280000/1 = 280000.0', '160000/1 = 160000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 2
['340000/2 = 170000.0', '280000/2 = 140000.0', '160000/1 = 160000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 3
['340000/3 = 113333.33333333333', '280000/2 = 140000.0', '160000/1 = 160000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 4
['340000/3 = 113333.33333333333', '280000/2 = 140000.0', '160000/2 = 80000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 5
['340000/3 = 113333.33333333333', '280000/3 = 93333.33333333333', '160000/2 = 80000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
Escanno: 6
['340000/4 = 85000.0', '280000/3 = 93333.33333333333', '160000/2 = 80000.0', '60000/1 = 60000.0', '15000/1 = 15000.0']
[3, 3, 1, 0, 0]
```

```
Escanno: 0
['391000/1 = 391000.0', '311000/1 = 311000.0', '184000/1 = 184000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 1
['391000/2 = 195500.0', '311000/1 = 311000.0', '184000/1 = 184000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 2
['391000/2 = 195500.0', '311000/2 = 155500.0', '184000/1 = 184000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 3
['391000/3 = 130333.33333333333', '311000/2 = 155500.0', '184000/1 = 184000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 4
['391000/3 = 130333.33333333333', '311000/2 = 155500.0', '184000/2 = 92000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 5
['391000/3 = 130333.33333333333', '311000/3 = 103666.66666666667', '184000/2 = 92000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 6
['391000/4 = 97750.0', '311000/3 = 103666.66666666667', '184000/2 = 92000.0', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
```

```

2000.0']
Escanno: 7
['391000/4 = 97750.0', '311000/4 = 77750.0', '184000/2 = 92000.0', '73000/1
= 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 8
['391000/5 = 78200.0', '311000/4 = 77750.0', '184000/2 = 92000.0', '73000/1
= 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 2000.0']
Escanno: 9
['391000/5 = 78200.0', '311000/4 = 77750.0', '184000/3 = 61333.3333333333
6', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 =
2000.0']
Escanno: 10
['391000/6 = 65166.666666666664', '311000/4 = 77750.0', '184000/3 = 61333.33
3333333336', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 11
['391000/6 = 65166.666666666664', '311000/5 = 62200.0', '184000/3 = 61333.33
3333333336', '73000/1 = 73000.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 12
['391000/6 = 65166.666666666664', '311000/5 = 62200.0', '184000/3 = 61333.33
3333333336', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 13
['391000/7 = 55857.142857142855', '311000/5 = 62200.0', '184000/3 = 61333.33
3333333336', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 14
['391000/7 = 55857.142857142855', '311000/6 = 51833.333333333336', '184000/3
= 61333.333333333336', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 =
12000.0', '2000/1 = 2000.0']
Escanno: 15
['391000/7 = 55857.142857142855', '311000/6 = 51833.333333333336', '184000/4
= 46000.0', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 16
['391000/8 = 48875.0', '311000/6 = 51833.333333333336', '184000/4 = 46000.
0', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 =
2000.0']
Escanno: 17
['391000/8 = 48875.0', '311000/7 = 44428.57142857143', '184000/4 = 46000.0',
'73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 = 200
0.0']
Escanno: 18
['391000/9 = 43444.444444444445', '311000/7 = 44428.57142857143', '184000/4
= 46000.0', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 19
['391000/9 = 43444.444444444445', '311000/7 = 44428.57142857143', '184000/5
= 36800.0', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0',
'2000/1 = 2000.0']
Escanno: 20
['391000/9 = 43444.444444444445', '311000/8 = 38875.0', '184000/5 = 36800.
0', '73000/2 = 36500.0', '27000/1 = 27000.0', '12000/1 = 12000.0', '2000/1 =
2000.0']
[9, 7, 4, 1, 0, 0, 0]

```

(d) Diagrama de barras

Si conociéramos los nombres de los partidos, con los números de los escaños podríamos hacer un bonito diagrama de barras, aunque sea a. Asumimos que los datos vienen dados en un diccionario que, para cada nombre de un partido, tiene aparejado su número de escaños.

En la demostración, observa que las longitudes se ajustan, de manera que el nombre más largo determina dónde aparecen las barras.

In [8]:



```
# Aquí viene la función definida por el estudiante
```

In [9]:



```
# Pruebas de funcionamiento:
```

```
trazar_diagrama_de_barras({"Los Verdes": 9, "Izquierdas": 17, "Norteños": 0, "Sur": 12, "Vi
```

```
Los Verdes          | *****
Izquierdas          | *****
Norteños            |
Sur                 | *****
Villa Arriba Ecológica | ****
```