Dada una mochila con una capacidad máxima de peso P, y dispones de n objetos. Cada objeto tiene un peso w y un valor v El objetivo es seleccionar algunos objetos de manera que maximicen el valor total, sin superar la capacidad de la mochila, que es de 10 Se pide realizar una prueba de escritorio para fuerza bruta y para programación dinámica (el valor máximo se encuentra en la celda dp[n][B], donde B es la capacidad máxima de la mochila)

**Datos de Entrada**

* **Capacidad máxima de la mochila (P):** 10
* **Número de objetos (n):** 4
* **Pesos (w) y valores (v) de los objetos:**
  + Objeto 1: w1=2*w*1​=2, v1=5*v*1​=5
  + Objeto 2: w2=3*w*2​=3, v2=8*v*2​=8
  + Objeto 3: w3=4*w*3​=4, v3=9*v*3​=9
  + Objeto 4: w4=5*w*4​=5, v4=10*v*4​=10

**1. Fuerza Bruta**

El enfoque de **fuerza bruta** consiste en evaluar todas las combinaciones posibles de objetos y seleccionar la que maximiza el valor sin exceder la capacidad de la mochila.

**Combinaciones posibles:**

1. **Sin objetos:**
   * Valor total: 0
   * Peso total: 0
2. **Objeto 1:**
   * Valor total: 5
   * Peso total: 2
3. **Objeto 2:**
   * Valor total: 8
   * Peso total: 3
4. **Objeto 3:**
   * Valor total: 9
   * Peso total: 4
5. **Objeto 4:**
   * Valor total: 10
   * Peso total: 5
6. **Objeto 1 + Objeto 2:**
   * Valor total: 5 + 8 = 13
   * Peso total: 2 + 3 = 5
7. **Objeto 1 + Objeto 3:**
   * Valor total: 5 + 9 = 14
   * Peso total: 2 + 4 = 6
8. **Objeto 1 + Objeto 4:**
   * Valor total: 5 + 10 = 15
   * Peso total: 2 + 5 = 7
9. **Objeto 2 + Objeto 3:**
   * Valor total: 8 + 9 = 17
   * Peso total: 3 + 4 = 7
10. **Objeto 2 + Objeto 4:**
    * Valor total: 8 + 10 = 18
    * Peso total: 3 + 5 = 8
11. **Objeto 3 + Objeto 4:**
    * Valor total: 9 + 10 = 19
    * Peso total: 4 + 5 = 9
12. **Objeto 1 + Objeto 2 + Objeto 3:**
    * Valor total: 5 + 8 + 9 = 22
    * Peso total: 2 + 3 + 4 = 9
13. **Objeto 1 + Objeto 2 + Objeto 4:**
    * Valor total: 5 + 8 + 10 = 23
    * Peso total: 2 + 3 + 5 = 10
14. **Objeto 1 + Objeto 3 + Objeto 4:**
    * Valor total: 5 + 9 + 10 = 24
    * Peso total: 2 + 4 + 5 = 11 (supera la capacidad, se descarta)
15. **Objeto 2 + Objeto 3 + Objeto 4:**
    * Valor total: 8 + 9 + 10 = 27
    * Peso total: 3 + 4 + 5 = 12 (supera la capacidad, se descarta)
16. **Objeto 1 + Objeto 2 + Objeto 3 + Objeto 4:**
    * Valor total: 5 + 8 + 9 + 10 = 32
    * Peso total: 2 + 3 + 4 + 5 = 14 (supera la capacidad, se descarta)

**Resultado de fuerza bruta:**

* La combinación óptima es **Objeto 1 + Objeto 2 + Objeto 4** con un valor total de **23** y un peso total de **10**.

**2. Programación Dinámica**

El enfoque de **programación dinámica** utiliza una tabla dp donde dp[i][j] representa el valor máximo que se puede obtener con los primeros i objetos y una capacidad de mochila j.

**Tabla dp:**

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 |  |  |  |  |  |  |  |  |  |  |
| 2 | 0 |  |  |  |  |  |  |  |  |  |  |
| 3 | 0 |  |  |  |  |  |  |  |  |  |  |
| 4 | 0 |  |  |  |  |  |  |  |  |  |  |

**Llenado de la tabla:**

1. **Fila 1 (Objeto 1):**
   * w1=2*w*1​=2, v1=5*v*1​=5
   * Para j≥2*j*≥2, dp[1][j]=5*dp*[1][*j*]=5.

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 0 |  |  |  |  |  |  |  |  |  |  |
| 3 | 0 |  |  |  |  |  |  |  |  |  |  |
| 4 | 0 |  |  |  |  |  |  |  |  |  |  |

1. **Fila 2 (Objeto 2):**
   * w2=3*w*2​=3, v2=8*v*2​=8
   * Para j≥3*j*≥3, dp[2][j]=max⁡(dp[1][j],dp[1][j−3]+8)*dp*[2][*j*]=max(*dp*[1][*j*],*dp*[1][*j*−3]+8).

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 0 | 0 | 5 | 8 | 8 | 13 | 13 | 13 | 13 | 13 | 13 |
| 3 | 0 |  |  |  |  |  |  |  |  |  |  |
| 4 | 0 |  |  |  |  |  |  |  |  |  |  |

1. **Fila 3 (Objeto 3):**
   * w3=4*w*3​=4, v3=9*v*3​=9
   * Para j≥4*j*≥4, dp[3][j]=max⁡(dp[2][j],dp[2][j−4]+9)*dp*[3][*j*]=max(*dp*[2][*j*],*dp*[2][*j*−4]+9).

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 0 | 0 | 5 | 8 | 8 | 13 | 13 | 13 | 13 | 13 | 13 |
| 3 | 0 | 0 | 5 | 8 | 9 | 13 | 14 | 17 | 17 | 17 | 22 |
| 4 | 0 |  |  |  |  |  |  |  |  |  |  |

1. **Fila 4 (Objeto 4):**
   * w4=5*w*4​=5, v4=10*v*4​=10
   * Para j≥5*j*≥5, dp[4][j]=max⁡(dp[3][j],dp[3][j−5]+10)*dp*[4][*j*]=max(*dp*[3][*j*],*dp*[3][*j*−5]+10).

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 0 | 0 | 5 | 8 | 8 | 13 | 13 | 13 | 13 | 13 | 13 |
| 3 | 0 | 0 | 5 | 8 | 9 | 13 | 14 | 17 | 17 | 17 | 22 |
| 4 | 0 | 0 | 5 | 8 | 9 | 13 | 14 | 17 | 18 | 19 | 23 |

**Resultado de programación dinámica:**

* El valor máximo se encuentra en dp[4][10]=23*dp*[4][10]=23.

**Comparación de Resultados**

* **Fuerza bruta:** Valor máximo = 23 (Objeto 1 + Objeto 2 + Objeto 4).
* **Programación dinámica:** Valor máximo = 23 (Objeto 1 + Objeto 2 + Objeto 4).

Ambos métodos coinciden en la solución óptima.