**Casos de prueba**

01\_CasoEnunciado

|  |  |
| --- | --- |
| 01\_CasoEnunciado.in | 01\_CasoEnunciado.out |
| 4 2  1 4  0 2 4 3  2 0 5 2  4 5 0 1  3 2 1 0 | 3  2 1  3 4 |

02\_UnaCiudadNecesitaElectricidad

Solo una ciudad necesita electricidad, se obtiene el costo mínimo

|  |  |
| --- | --- |
| 02\_UnaCiudadNecesitaElectricidad.in | 02\_UnaCiudadNecesitaElectricidad.out |
| 4 3  2 3 4  0 3 5 6  3 0 2 4  5 2 0 1  6 4 1 0 | 3  2 1 |

03\_TendidoElectricoDeMenorCosto

Se comprueba que, efectivamente, el programa elija el menor costo.

|  |  |
| --- | --- |
| 03\_TendidoElectricoDeMenorCosto.in | 03\_TendidoElectricoDeMenorCosto.out |
| 4 2  1 4  0 1 4 3  1 0 5 2  4 5 0 1  3 2 1 0 | 2  2 1  3 4 |

**Consideraciones:**

* Si existe de una ciudad dos centrales eléctricas con igual costo para abastecerse tomará la primera aparición, ejemplo: en el caso del enunciado se puede elegir la arista 2 1 cuyo costo es 2 siendo 1 una central eléctrica o se puede elegir la arista 2 4 cuyo costo también es 2 y 4 también es una central eléctrica. Es aquí cuando el algoritmo elige la arista 2 1.
* **Se obtiene como resultado solo el costo ya que eso pide el ejercicio, no un camino.**
* Complejidad Computacional: O(N^3)
* Algoritmo usado: Floyd