

**Descripción.** Usted trabaja en una compañía que a diario debe realizar el proceso de entrega de sus productos. Para lograr este propósito, un camión se carga completamente al inicio del día, saliendo del depósito (home) y recorriendo múltiples puntos de la ciudad para finalmente volver al punto inicial. El director logístico del proyecto está interesado en **realizar mejoras a la ruta actual** con el fin de **reducir la distancia total recorrida**, y por tanto, **el combustible** gastado durante la totalidad de la entrega. Para esto, a usted le han solicitado que realice la siguiente implementación:

**Requerimiento.** Escriba una función (*ruteo*) que reciba como parámetros: **Un diccionario (*distancias*) en el cuál para una llave (i, j) el valor correspondiente es la distancia que hay entre el lugar *i* y el lugar *j*. Y una lista (*ruta\_inicial*) la cual contiene el orden de visita de los distintos puntos de la ciudad según la ruta actual.** Con esta información, realice el procedimiento que se enuncia a continuación:

En cada iteración usted debe evaluar todos los posibles **intercambios de dos paradas** en la ruta para finalmente actualizarla con aquel intercambio que genere la **mayor** reducción en la distancia total recorrida. **Una vez actualizada la ruta, debe pasar a la siguiente iteración y repetir completamente este proceso hasta que no se encuentren mejoras adicionales (criterio de parada).** Entonces, retorne un diccionario con las llaves “ruta” y “distancia” donde, el valor correspondiente a la primera llave sea una cadena que codifique la ruta final encontrada con las paradas separadas por guiones y el valor correspondiente a la segunda llave sea un entero con la distancia total asociada a la ruta reportada. **Note qué** Debido a la configuración vial, la distancia que se debe recorrer para ir en automóvil de un lugar i a un lugar j no necesariamente será la misma distancia que se debe recorrer para ir del lugar j al lugar i (**ida y vuelta no son equivalentes**). Independientemente de esto, **la ruta siempre debe iniciar y terminar el depósito.**

**Esqueleto.**

```
def ruteo(distancias: dict, ruta_inicial: list)-> dict:
    pass
```

**Pruebas públicas.**

Entradas		Salida
distancias	ruta_inicial	return
{('H', 'H'): 0, ('H', 'A'): 21, ('H', 'B'): 57, ('H', 'C'): 58, ('H', 'D'): 195, ('H', 'E'): 245, ('H', 'F'): 241, ('A', 'H'): 127, ('A', 'A'): 0, ('A', 'B'): 231, ('A', 'C'): 113, ('A', 'D'): 254, ('A', 'E'): 179, ('A', 'F'): 41, ('B', 'H'): 153, ('B', 'A'): 252, ('B', 'B'): 0, ('B', 'C'): 56, ('B', 'D'): 126, ('B', 'E'): 160, ('B', 'F'): 269, ('C', 'H'): 196, ('C', 'A'): 128, ('C', 'B'): 80, ('C', 'C'): 0, ('C', 'D'): 136, ('C', 'E'): 37, ('C', 'F'): 180, ('D', 'H'): 30, ('D', 'A'): 40, ('D', 'B'): 256, ('D', 'C'): 121, ('D', 'D'): 0, ('D', 'E'): 194, ('D', 'F'): 109, ('E', 'H'): 33, ('E', 'A'): 144, ('E', 'B'): 179, ('E', 'C'): 114, ('E', 'D'): 237, ('E', 'E'): 0, ('E', 'F'): 119, ('F', 'H'): 267, ('F', 'A'): 61, ('F', 'B'): 79, ('F', 'C'): 39, ('F', 'D'): 135, ('F', 'E'): 55, ('F', 'F'): 0}	['H', 'A', 'B', 'C', 'D', 'E', 'F', 'H']	{'ruta': 'H-A-F-B-D-C-E-H', 'distancia': 458}
{('H', 'H'): 0, ('H', 'A'): 60, ('H', 'B'): 202, ('H', 'C'): 206, ('H', 'D'): 40, ('H', 'E'): 27, ('A', 'H'): 72, ('A', 'A'): 0, ('A', 'B'): 135, ('A', 'C'): 150, ('A', 'D'): 240, ('A', 'E'): 117, ('B', 'H'): 188, ('B', 'A'): 166, ('B', 'B'): 0, ('B', 'C'): 149, ('B', 'D'): 126, ('B', 'E'): 199, ('C', 'H'): 39, ('C', 'A'): 19, ('C', 'B'): 123, ('C', 'C'): 0, ('C', 'D'): 206, ('C', 'E'): 19, ('D', 'H'): 45, ('D', 'A'): 14, ('D', 'B'): 110, ('D', 'C'): 95, ('D', 'D'): 0, ('D', 'E'): 31, ('E', 'H'): 36, ('E', 'A'): 179, ('E', 'B'): 235, ('E', 'C'): 106, ('E', 'D'): 25, ('E', 'E'): 0}	['H', 'B', 'E', 'A', 'C', 'D', 'H']	{'ruta': 'H-D-A-B-C-E-H', 'distancia': 393}

**Ejemplo comparaciones.** Suponga que para una iteración cualquiera usted cuenta con la ruta [‘H’, ‘A’, ‘B’, ‘C’, ‘D’, ‘H’]. Entonces, todas las posibles parejas de intercambio que se deberían evaluar son las siguientes: (‘A’, ‘B’); (‘A’, ‘C’); (‘A’, ‘D’); (‘B’, ‘C’); (‘B’, ‘D’), (‘C’, ‘D’). Si por ejemplo, usted encontrase que entre estos, el mejor intercambio corresponde a la pareja (‘B’, ‘D’), la ruta actualizada para la siguiente iteración deberá ser [‘H’, ‘A’, ‘D’, ‘C’, ‘B’, ‘H’] y deberá nuevamente encontrar y evaluar todas las posibles parejas las cuales en este caso serían: (‘A’, ‘D’); (‘A’, ‘C’); (‘A’, ‘B’); (‘D’, ‘C’); (‘D’, ‘B’); (‘C’, ‘B’). Pensar en el orden de recorrido o enunciación de las parejas de este ejemplo le puede ayudar a identificar cómo plantear las comparaciones de forma genérica en su código.

**Validación.** Para este reto, usted debe verificar que **todas las distancias del diccionario que entra por parámetro sean positivas ó cero en el caso de las distancias entre dos lugares iguales.** En cualquier otro caso retorne una **cadena** reportando el siguiente mensaje de error “Por favor revisar los datos de entrada.”