

Diario de Motocicleta

Examen Extraordinario de Programación - Curso 2024

Día 1



Advertencia

Si usted está leyendo este documento si haberlo descomprimido, ciérrelo inmediatamente y descomprima el contenido del archivo zip que recibió en el escritorio. Una vez hecho esto, reanude su trabajo desde allí. Trabajar antes de descomprimir le hará perder sus cambios, y no tendrá oportunidad de reclamar.

La empresa *Gafitas Para Ti* está muy contenta con el trabajo realizado por Noel para acabar con la empresa rival *Dile No A La Miopía*. Su trabajo fue de tal calidad que decidieron cambiarlo de provincia y le asignaron toda una sucursal para que la manejase, vamos lo que es un gerente de toda la vida. Incluso le dieron una motocicleta Ural de 1967, que casi todos los días funciona, para que haga las reparticiones.

Noel sabe que todo el mérito no es suyo, le debe un gran favor a los estudiantes de Ciencias de la Computación por tanto decide contratarlos a tiempo parcial para que sean el equipo de logística y planificación de los repartos.

La Provincia de Gafalandia donde vive Noel ahora presenta una crisis aguda con el combustible, vaya por dios, por lo que solo hay combustible en una de las ciudades, justo donde está la sucursal de la empresa. Noel quiere hacer un buen trabajo en su nueva empresa y para eso desea lograr atender TODAS las demandas de gafas recibidas. Como la moto tiene *sidecar*, afortunadamente Noel puede cargar todas las cajas de gafas que quiera, el problema está en el consumo de combustible.

De manera general será casi imposible pasar por todas las ciudades en un solo viaje. Como solo hay combustible en la ciudad origen, Noel deberá regresar a esta ciudad en varias ocasiones para rellenar el tanque.

Le informa de la situación a su equipo de logística y estos le dicen que encontraran una forma de ir desde la ciudad que tiene combustible en estos momentos y visitar el resto de las ciudades gastando la menor cantidad de combustible posible. Debido a que la capacidad de combustible del transporte es limitada, es probable que Noel tenga que regresar a la ciudad origen varias ocasiones para rellenar el tanque.

Descripción del Problema

El problema consiste en encontrar la ruta más eficiente para visitar todas las ciudades partiendo desde una ciudad origen cualquiera. Esta ruta siempre comienza y termina en la ciudad de origen, y visita cada ciudad (excepto el origen) exactamente una vez.

Se conoce el costo de ir desde una ciudad i hasta una ciudad j dado por una matriz `int[,]` `map` de enteros la cual no tiene que ser necesariamente simétrica, es decir que puede ocurrir que `map[i,j] != map[j,i]`. Además se conoce la capacidad máxima de combustible que puede usar la única y destartalada motocicleta disponible que se usará para el reparto.

Nota

Este problema es conocido por ser NP-hard, lo que significa que muy probablemente no haya solución eficiente. Concéntrese en implementar un algoritmo de búsqueda exhaustiva que explore todas las posibles rutas.

Entrada

1. Una matriz **map** de $N \times N$ enteros donde $map_{i,j}$ es el costo de ir desde la ciudad i hasta la ciudad j .
2. Un entero **capacity** que representa la máxima capacidad de combustible que puede almacenar la motocicleta del repartidor.
3. Un entero **origin** que representa la ciudad de la que se desea comenzar el reparto de gafas, y a donde se puede regresar tantas veces como sea necesario para rellenar el tanque.

Salida

El costo total de combustible mínimo necesario para visitar todas las ciudades exactamente una vez, regresando al origen tantas veces como sea necesario para rellenar el tanque, y terminando en el origen. En caso de no ser posible visitar todas las ciudades, devolver -1 .

Implementación

Usted debe entregar un solo archivo **Nombre-Apellido1-Apellido2-C1XX.cs** con su implementación del siguiente método. Asegúrese de incluir todo el código adicional que necesite para que su solución funcione en la clase **Solution**. No cambie la signatura del método **Solve**.

```
namespace Weboo.Examen
{
    public static class Solution
    {
        public static int Solve(int[,] map, int capacity, int origin)
        {
            throw new NotImplementedException();
        }
    }
}
```

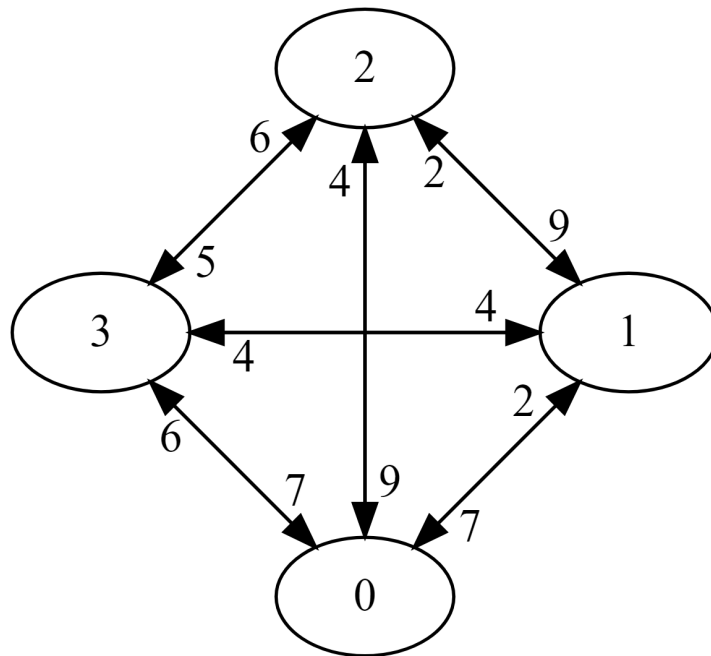
Notas

1. Los valores de **map** serán siempre mayores estrictos que 0 entre todos los pares de ciudades i, j . Es decir, si $i \neq j$ siempre será posible ir de una ciudad i a una ciudad j y viceversa, y el costo de combustible será siempre mayor a 0.
2. Para todo trío de ciudades i, j, k distintas, se cumple que $map[i, j] + map[j, k] \geq map[i, k]$.
3. Una ruta válida siempre comienza y termina en la ciudad de origen.

4. El valor de **capacity** siempre será un entero positivo.
5. El tanque de combustible comienza lleno, y regresar a la ciudad **origin** implica rellenarlo hasta el tope (**capacity**).
6. Es imprescindible visitar cada ciudad *exactamente una vez*, excepto la ciudad origen, que puede visitarse tantas veces como sea necesario para rellenar el tanque.
7. Puede existir más de una solución óptima, o ninguna solución.
8. El costo total del viaje es la suma de los costos de todos los desplazamientos que haga Noel, incluyendo los viajes de regreso a rellenar combustible. No importa si queda combustible en el tanque al terminar, solo cuenta el que se gastó.

Ejemplo 1

Considere la siguiente configuración de entrada, donde la ciudad de origen es 1, y en cada arista, el valor más cercano a la flecha define el costo de ir hacia dicha ciudad. Por ejemplo, el costo de ir desde 1 hasta 0 es 7 litros de combustible, mientras que el costo de regresar de 0 a 1 es 2 litros.

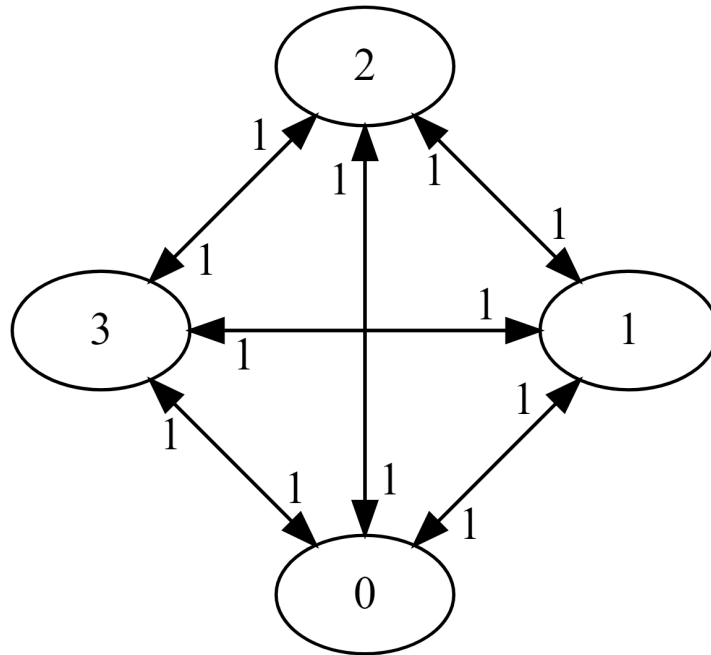


Consideremos que tenemos un tanque con 14 litros de capacidad. Una ruta óptima sería desde la ciudad 1, ir a la ciudad 2, luego a la ciudad 3, luego regresar a la ciudad 1 a rellenar el combustible, después a la ciudad 0, y finalmente regresar a la ciudad 1.

Costo total: 20 unidades de combustible.

Ejemplo 2

Analicemos el siguiente mapa, tomando como ciudad origen 0, usando diferentes capacidades de tanque.



1. capacity = 1 No es posible visitar todas las ciudades con una capacidad de tanque de 1, pues si sales del origen, luego no podrías volver a rellenar combustible. Por lo tanto, el resultado esperado es -1.
2. capacity = 2
 - Una ruta óptima sería desde la ciudad 0, ir a la ciudad 1, luego regresar a 0 a recargar, luego ir a la ciudad 2, después regresar a la ciudad 0 para rellenar el tanque, después ir a la ciudad 3 y finalmente regresar a la ciudad 0.
 - **Costo total:** 6 unidades de combustible.
3. capacity = 3
 - Una ruta óptima sería desde la ciudad 0, ir a la ciudad 1, luego a la ciudad 2, luego regresar a la ciudad 0 para rellenar el tanque, después visitar la ciudad 3, y finalmente regresar a la ciudad 0.
 - **Costo total:** 5 unidades de combustible.
4. capacity = 4

- Una ruta óptima sería desde la ciudad 0, ir a la ciudad 1, luego a la ciudad 2, después a la ciudad 3, y finalmente regresar a la ciudad 0.
- **Costo total:** 4 unidades de combustible.