

Segundo Examen de Programación

Curso 2024



Figure 1: Noel

En el país Gafa-Landia, todas las personas usan gafas y bueno estas de vez en cuando necesitan ser cambiadas o reparadas. Para ello la empresa “Gafitas Para Ti” trabaja arduamente en complacer a sus clientes. Dicha industria cuenta con un servicio de reparto de gafas, el cual se extiende por todo el país. Sin embargo una empresa nueva “Dile No A La Miopía” acaba de ser inaugurada y amenaza con arrebatar y destruir la reputación de “Gafitas Para Ti”. Noel es uno de los empleados de esta última y al igual que todos sus compañeros, está dispuesto a defender su centro de trabajo

Para que la empresa enemiga no se salga con la suya, “Gafitas Para Ti”, decidió responder a más pedidos que “Dile No A La Miopía”. Cada ciudad de Gafa-Landia tiene una cantidad de pedidos que deben ser atendidos, los cuales se representan en un array de enteros **GlassForCitys**, donde la posición i representa la cantidad de pedidos que deben ser complacidos en la i -ésima ciudad. Además se conoce cuanto combustible se consume en ir desde una ciudad a la otra (Map) y el combustible máximo que puede ser gastado por Noel (MaxGasol).

Noel después de ver estos datos se dio cuenta que **NO** siempre podrá complacer **TODAS** las demandas, así que intentará complacer la **Mayor Cantidad Posible**, de esta manera el índice de aceptación de su empresa superará al del rival. Por eso contrató los servicios de un profesional (o sea tú) para que le ayudes

a encontrar un recorrido válido que le permita obtener la cantidad máxima de demandas que podrá complacer conociendo los datos anteriores. Llamemos recorrido válido a todo recorrido que comience y termine en **GlassForCitys[0]** y consuma una cantidad de combustible menor o igual que **MaxGasol**

Entrada

- **GlassForCitys** array de enteros de longitud n
- **MaxGasol** entero que representa la máxima cantidad de combustible que se puede gastar en un recorrido
- **Map** matriz de enteros...(se explicará en breve)

Salida

Usted debe retornar un array de enteros que represente un recorrido válido que maximice la cantidad de demandas que puede complacer Noel. Para ello implemente el método **Solve** de la clase **Solution** en el archivo *Solution.cs*.

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

Map

Map es una matriz de enteros, simétrica es decir que $Map[i, j] = Map[j, i]$. Cada celda de la matriz representa el costo de ir desde la ciudad i hasta la ciudad j . Si no es posible ir desde una ciudad a otra entonces, se cumplirá que: $Map[i, j] = -1$. Además $Map[i, i] = 0, \forall i$.

Aclaraciones

1. Los valores de entrada nunca serán *null* y todos los valores numéricos siempre serán mayores o iguales que 0, excepto algún valor de Map.
2. Un recorrido válido es aquel que empieza y termina en **GlassForCitys[0]**

3. Se puede visitar una misma ciudad tantas veces se considere necesario: Sin embargo el reparto de gafas solo se puede realizar una vez, aunque siempre se consumirá combustible, independientemente de si la ciudad fue visitada o no.
4. Asuma que una vez que Noel llega a la ciudad, se baja de la moto y complace **TODAS** las demandas de la misma a pie.
5. No puede utilizar en su solución la clase *GafasUtils* que distribuye en un ensamblado junto al proyecto.

Ejemplos

$$GlassForCitys = \{0, 32, 10, 43\}$$

$$MaxGasol = 49$$

$$Map = \begin{pmatrix} 0 & 10 & -1 & 10 \\ 10 & 0 & 15 & -1 \\ -1 & 15 & 0 & -1 \\ 10 & -1 & -1 & 0 \end{pmatrix}$$

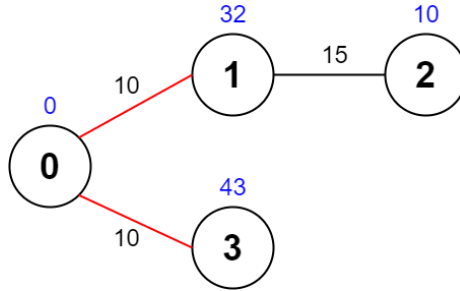


Figure 2: Ejemplo 1

Máxima de demanda que puede satisfacer: 75. En este caso un posible recorrido sería $\{0, 1, 0, 3, 0\}$. En este se visitan las ciudades 0,1 y 3 por tanto la cantidad de entregas realizadas sería: $0 + 32 + 43 = 75$. Note que en el recorrido $\{0, 1, 2, 1, 0\}$ se realizan más entregas, pero el tiempo consumido es mayor que *MaxGasol*.

$$GlassForCitys = \{5, 10, 15, 20\}$$

$$MaxGasol = 30$$

$$Map = \begin{pmatrix} 0 & 10 & -1 & 10 \\ 10 & 0 & 10 & -1 \\ -1 & 10 & 0 & -1 \\ 10 & -1 & -1 & 0 \end{pmatrix}$$

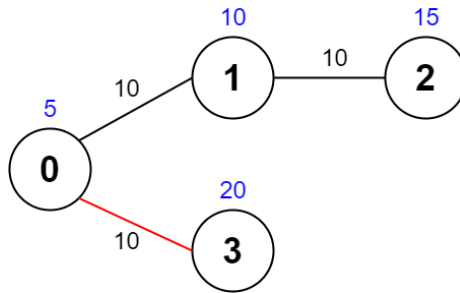


Figure 3: Ejemplo 2

Máxima de demanda que puede satisfacer: 25. En este caso un posible recorrido sería $\{0, 3, 0\}$. En este se visitan las ciudades 0 y 3 por tanto la cantidad de entregas realizadas sería: $5 + 20 = 25$. Note que en el recorrido $\{0, 1, 0\}$ se consume la misma cantidad de tiempo, pero la cantidad de entregas realizadas no es la máxima.

$$GlassForCitys = \{1, 2, 3, 4\}$$

$$MaxGasol = 50$$

$$Map = \begin{pmatrix} 0 & 10 & -1 & -1 \\ 10 & 0 & 11 & 13 \\ -1 & 11 & 0 & 12 \\ -1 & 13 & 12 & 0 \end{pmatrix}$$

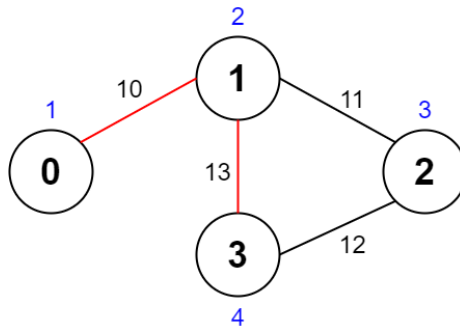


Figure 4: Ejemplo 3

Máxima de demanda que puede satisfacer: 7. En este caso un posible recorrido sería $\{0, 1, 3, 1, 0\}$. En este se visitan las ciudades 0,1 y 3 por tanto la cantidad de entregas realizadas sería: $1 + 2 + 4 = 7$.