



Laboratorio Presencial - Algoritmos avanzados

Integrantes: Cristian Espinoza
Curso: Algoritmos Avanzados
Sección: 0-A-1
Profesor(a): Mónica Villanueva
Ayudante(s): Nicolás Gutierrez

24 de Diciembre de 2018

Tabla de contenidos

1. Enunciado 1	1
1.1. Problema	1
1.2. Idea	1
1.3. Algoritmo	1
1.4. Traza	3

1. Enunciado 1

1.1. Problema

Realizar una idea, algoritmo y traza de la posible solución que le daremos al problema planteado.

1.2. Idea

La idea que planteamos en esta ocasión para poder ayudar a la empresa urbana de limpieza, es en primera instancia leer el archivo de entrada que contendrá el número de centros de acopio que existen, la cantidad de incineradores que están operativos y la cantidad en millones de pesos que entrega el gobierno como subsidio y por último, también leer cada una de la distancia que contiene el centro de acopio a la ciudad y la cantidad de tonelada que conserva dentro de él. Luego de esto abarcaremos el problema de la siguiente forma:

- 1. Guardaremos la información de cada uno de los centros en una estructura llamada Nodo, que en la sección de algoritmo se verá su implementación, pero en simples palabras, contendrá la distancia que posee el nodo de ir a la ciudad y la cantidad de tonelada que conserva en él.
- 2. Realizaremos la obtención de costos de cada uno de los traslados que se puede realizar de un nodo hacia todos los otros nodos, mediante realizamos este cálculo vamos guardando el que tenga el costo menor con el fin de realizar el traslado de la basura al nuevo centro de acopio.
- 3. Luego, el paso anterior lo repetimos hasta que se consiga dejar la basura en la cantidad de centros de acopio habilitados (Dato que es sacado desde el archivo de entrada). Teniendo en cuenta que solo vamos a considerar los centros de acopio que contiene basura dentro de él, es decir, que no se consideran los que contienen 0 toneladas. Almacenando el destino nuevo y el costo que contiene cada traslado que se está realizando y por último, ir actualizando la nueva cantidad de toneladas que va a contener ese centro de acopio.
- 4. Por último, realizamos la escritura de archivo donde vamos a colocar los traslados realizados para llegar a la solución, además de mostrar cuáles fueron los centros de acopio que se quedaron con toneladas de basura y cuánto tiene respectivamente, y por último indicar el costo total que se tuvo al realizar cada uno de los traslados hacia los nuevos centros de acopio.

1.3. Algoritmo

```
Struct Nodo {
    int distanceOrigin;
    int weight;
}

Struct Lista {
    int numbersSCollectionCenter;
    int numbersSCollectionAvailable;
    int governmentMoney;
    int cost;
    Nodo* nodo:
}
```

```

Goloso (Lista* listNodos)
sumTotal;
auxSum;
indexOrigin;
indexDestination;
bandera = 0;
counter = 0;

Mientras counter < listNodos.numbersSCollectionAvailable

    Para cada nodo en listNodos.nodo:

        # Es la parte que se mantiene constante del calculo de costo.
        x = listNodos->nodo.weight - (listNodos->nodo.weight / listNodos.governmentMoney)

        # No consideramos centros con tonelada 0 de basura
        Si nodo.weight != 0 :

            Para cada nodoAux en listNodos.nodo:
                Si ( nodo distinto de nodoAux )
                    Si ( bandera igual a 0 )
                        sumTotal = nodo.distanceOrigin - nodoAux.distanceOrigin * x;
                        bandera = 1;
                        indexOrigin = nodo;
                        indexDestination = nodoAux;
                    Si no
                        auxSum = nodo.distanceOrigin - nodoAux.distanceOrigin * x;
                        Si (suma >= auxSum)
                            suma = auxSum;
                            indexOrigin = nodo;
                            indexDestination = nodoAux;

            sumTotal = 0;
            auxSum = 0;
            bandera = 0;

        # Realizamos la escritura del movimiento que se realizo
        escribirArchivo(indexOrigin,indexDestination);

        # Actualizamos el costo de los nuevos centros y
        # cuales quedaron sin basura en su interior
        actualizarCostos(indexOrigin, indexDestination, cost, counter);

# Luego escribimos en el archivo los centros de acopio que quedaron
# y con la basura respectiva que contiene. Por ultimo, agregamos al
# archivo el costo total que se utilizo para realizar las
# distintas trasferencias de basura de un centro a otro.
escribirFinalArchivo(listNodos)

```

1.4. Traza

Realizaremos la traza con el ejemplo subido al enunciado numero 1, el cual la entrada de ese archivo consiste en lo siguiente:

Entrada1.in
4 2 5
5 20
14 35
24 17
31 2

Figura 1: Representación gráfica de la entrada

Comenzamos aplicar el algoritmo planteado anteriormente:

```
# Leemos el archivo y almacenamos los datos para poder comenzar con la aplicaci n del algoritmo plan

# Primera iteraci n:

# Nodo: 17 toneladas - 24 distancia al origen
x = (17 - (17 / 5) ) = 13.6
24 - 5 * x = 258
24 - 31 * x = 95.2
24 - 14 * x = 136

# Nodo: 20 toneladas - 5 distancia al origen
x = (20 - (20 / 5) ) = 16
20 - 24 * x = 304
20 - 31 * x = 416
20 - 14 * x = 144

# Nodo: 35 toneladas - 14 distancia al origen
x = (35 - (35 / 5) ) = 28
14 - 5 * x = 252
14 - 24 * x = 280
14 - 31 * x = 144

# Nodo: 2 toneladas - 31 distancia al origen
x = (2 - (2 / 5) ) = 1.6
31 - 5 * x = 41.6
31 - 24 * x = 11.2
31 - 14 * x = 27.2

# Luego que calculamos cada una de los costos que se realiza de un nodo hacia todo los dem s ,
# llegamos que el que contiene el costo menor es la transferencia de 31 -> 24.

# Luego escribimos esa transferencia de tonelada en el archivo y actualizamos los
# costos que contiene cada centro.

# Quedando el nodo que tiene una distancia de 24 al origen con 19 toneladas al
# recibir las toneladas del centro 31.

# Aumentamos la variable de costo, debido a que realizamos una transferencia que
# costo un costo en especifico, tomando un valor de 11.2.

# Aumentamos la variable que mantiene informado de cuantos centros de acopio han
# quedado con 0 toneladas.

# Segunda iteraci n:

# Nodo: 19 toneladas - 24 distancia al origen
x = (19 - (19 / 5) ) = 15.2
24 - 5 * x = 288.8
24 - 14 * x = 152

# Nodo: 20 toneladas - 5 distancia al origen
x = (20 - (20 / 5) ) = 16
20 - 24 * x = 304
20 - 14 * x = 144
```

```

# Nodo: 35 toneladas - 14 distancia al origen
x = (35 - (35 / 5) ) = 28
14 - 5 * x = 252
14 - 24 * x = 280

# Luego que calculamos cada una de los costos que se realiza de un nodo hacia todo
# los dem s , llegamos que el que contiene el costo menor es la transferencia de 5 -> 14.

# Luego escribimos esa transferencia de tonelada en el archivo y actualizamos
# los costos que contiene cada centro.

# Quedando el nodo que tiene una distancia de 14 al origen con 55 toneladas
# al recibir las toneladas del centro 5.

# Aumentamos la variable de costo, debido a que realizamos una transferencia
# que costo un costo en especifico, el cual varia al costo de 155.2.

# Aumentamos la variable que mantiene informado de cuantos centros de acopio
# han quedado con 0 toneladas.

# Luego tenemos que el contador de centros con 0 toneladas, iguala a la cantidad de
# centros disponibles con que se cuenta (Dato le do desde el archivo de entrada),
# as teniendo en cuenta que se termina de realizar el ciclo.

```

Por ultimo, podemos decir que lo entregado al realizar la traza de nuestro algoritmo coincide con el archivo de salida que esta indicado en el enunciado numero 1. En donde tenemos que los movimientos realizados son del centro 31 ->24 y 5 ->14, utilizando un costo total de 155.2 y por ultimo. indicando que tenemos que el centro 14 queda con un total de 55 toneladas y el nodo 24 con 19 toneladas.

Salida1.out
31->24 5->14 centro 14: 55 toneladas centro 24: 19 toneladas Costo: 155.2

Figura 2: Representación gráfica de la entrada