UNIVERSIDAD DE CARABOBO
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
U.A. de Algoritmos, Programación y Lenguajes
CAO403: Programación II

Taller #3
Período Lectivo 1-2017
(Valor: 30%)

1.INSTRUCCIONES

- Este taller de programación es una evaluación estrictamente individual; por lo tanto, durante la realización del mismo, no está permitido prestar o recibir ayuda de otro(s) estudiante(s).
- ESTA permitido el uso de códigos fuentes en digital hechos por usted para el desarrollo del taller.
- Pueden traer material bibliográfico (libros, guías, manuales, código impreso, etc.) para el desarrollo del taller de programación.
- El estudiante que infrinja alguna de las instrucciones anteriores será retirado del taller y se le considerará aplazado con la nota mínima (Artículo 34 de las Normas de Evaluación de los Aprendizajes).
- **El taller tiene una duración estimada de tres (03) horas académicas.**
- El ejercicio propuesto requiere la lectura de datos de la entrada estándar (*standard input*) y la escritura de resultados en la salida estándar (*standard output*).
- Para el desarrollo del taller debe utilizar el lenguaje de programación C++, junto con sus librerías estándar.
- Debe proveer un archivo ***makefile*** que permita compilar su taller, la falta del mismo conllevará a la no revisión de su taller, **adicionalmente dicho *makefile* debe generar un ejecutable de nombre "*prim*",** con el fin de facilitar las labores de corrección de su taller, el no hacerlo generara puntos menos en su nota final del taller.

2.ACTIVIDADES A REALIZAR

El **algoritmo de *Prim*** es un algoritmo de la teoría de grafos para encontrar un árbol de expansión mínimo en un grafo conexo, **no dirigido** y cuyas aristas están etiquetadas. En otras palabras, el algoritmo encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo, entonces el algoritmo encontrará el árbol de expansión mínimo para uno de los componentes conexos que forman dicho grafo no conexo.

Pseudo código del algoritmo

La idea básica consiste en añadir, en cada paso, una arista de peso mínimo a un árbol previamente construido. Más explícitamente:

Sean:

$G = (V, A)$ un grafo grafo pesado conexo, no dirigido.
T : Conjunto[Arista]
B : Conjunto[Vértice]
u, v : Vértice

Paso 1. Se elige un vértice $\boldsymbol{u \in V}$ y se considera el árbol $\boldsymbol{B = \{u\}}$ y $\boldsymbol{T = \varnothing}$

Paso 2. Se considera la arista $\boldsymbol{\{u, v\} \in A}$ de mínimo peso que une un vértice $\boldsymbol{u \in B}$ y un vértice $\boldsymbol{v \in (V - B)}$, y se hace $\boldsymbol{B \leftarrow B \cup \{ v \}}$ y $\boldsymbol{T \leftarrow T \cup \{ \{u, v\} \}}$.
Paso 3. Si $\boldsymbol{|B| = |V|}$, el algoritmo termina. En caso contrario se vuelve al **paso 2**

2.1.FORMATO DE ENTRADA

Cada caso de prueba constará de líneas con ternas ***a***, ***b*** y ***c***, siendo ***a*** y ***b*** de tipo ***string*** y c de tipo ***float*** ($0 < c < 1000$), que representan una arista ***(a, b)*** con peso **c**, perteneciente al grafo. Cada caso de prueba terminará con la terna $\boldsymbol{a = *}$, $\boldsymbol{b = *}$ y $\boldsymbol{c = 0.0}$, la cual no forma parte del grafo.

Ejemplo

```
A B 7.0
A D 5.0
B C 8.0
B D 9.0
B E 7.0
C E 5.0
D E 15.0
D F 6.0
E F 8.0
E G 8.0
F G 11.0
* * 0.0
```

2.2.FORMATO DE SALIDA

Para cada caso de prueba se debe imprimir su árbol cobertor de costo mínimo, con el costo total del mismo, cumpliendo el formato que se muestra a continuación. Cada caso de prueba termina con una línea en blanco.

Ejemplo.

```
Caso #1:
A B 7.0
A D 5.0
B E 7.0
C E 5.0
D F 6.0
E G 8.0
Total: 38.0
```

3.OBSERVACIONES

- No debe mostrar por pantalla ningún tipo de mensaje al usuario para su interacción con el programa, ninguna interfaz gráfica desarrollada por usted, será corregida positivamente o le aportara ningún punto extra para su calificación final
- El formato de salida debe ser respetado como se indica en el enunciado, de lo contrario acarreará con puntos menos en su nota final.