

# Programario: Unidad 2 - Tema 1

Said Polanco-Martagón

April 14, 2017

## 1 Instrucciones

Siga las instrucciones descritas en el **Programario 1**.

## 2 Problemas

1. Considere la Figura 1.
  - (a) Genere un archivo CSV con la matriz de adyacencia del grafo de la Figura 1.
  - (b) Genere un archivo CSV con la matriz de incidencia del grafo de la Figura 1.
2. Tomando las matrices generadas en el problema 1 realice lo siguiente:
  - (a) Implemente el algoritmo de PRIM para generar el árbol de recubrimiento mínimo y guárdelo como una matriz en un archivo CSV.
  - (b) Implemente el algoritmo de Kruskal para generar el árbol de recubrimiento mínimo y guárdelo como una matriz en un archivo CSV.
  - (c) ¿Qué algoritmo tiene mejor desempeño en tiempo?
3. Implemente los algoritmos de: Dijkstra, Floyd y Bellman-ford para crear una aplicación que obtenga el camino más corto de un punto a otro de la Figura 1. La aplicación deberá permitir seleccionar el algoritmo a utilizar.
4. Considere el grafo de la Figura 1. Implemente una aplicación que permita saber si un grafo es sub-grafo de otro.

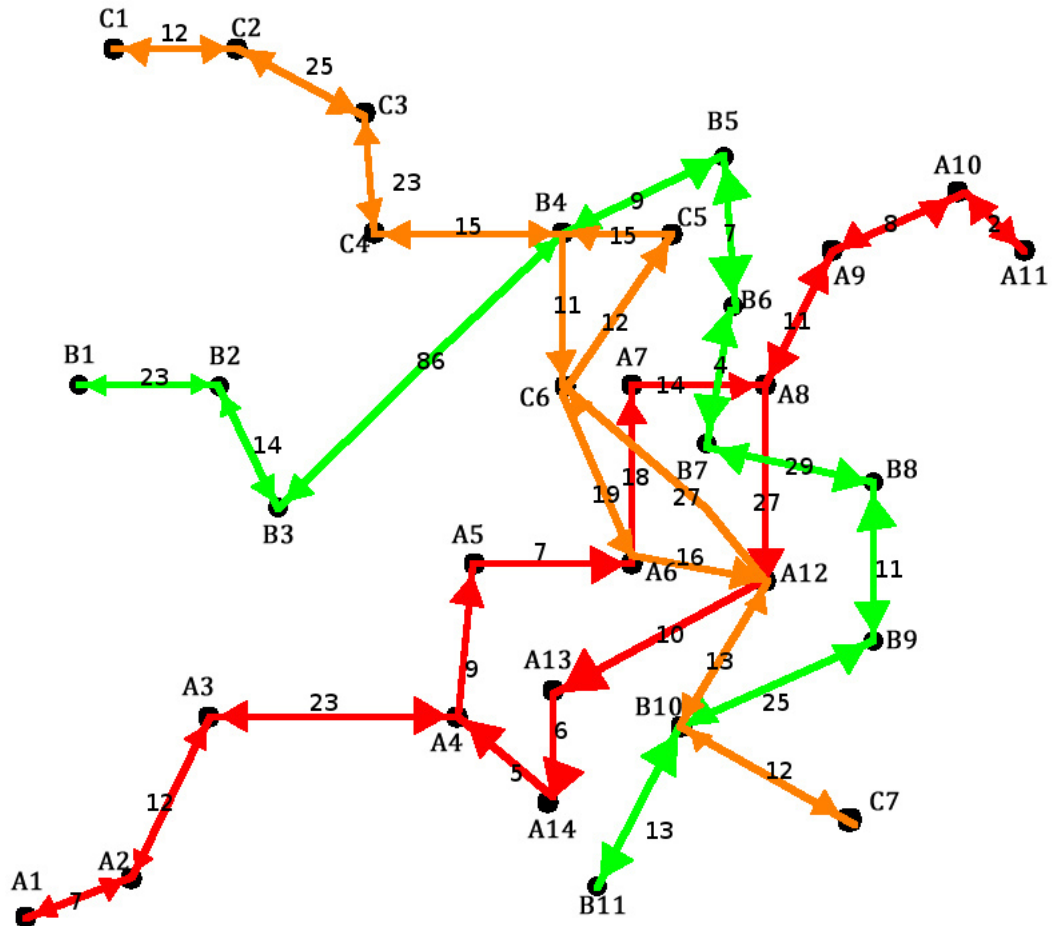


Figure 1: Rutas

- (a) Archivo del grafo original. Si fuese necesario podrá contener la matriz de adyacencia y de incidencia.
- (b) Archivo del grafo a comparar. Si fuese necesario podrá contener la matriz de adyacencia y de incidencia.

5. Mediante plantillas, recursión y listas enlazadas, implemente un árbol

binario de búsqueda.

6. Implemente un árbol binario de expresión para interpretar ecuaciones matemáticas que contengan:  $+$ ,  $-$ ,  $*$ ,  $/$ , *sqrt*, *sin*, *cos*, *tanh*, *cubo*, *exp*, *logn*. Y además, que permita la graficación mediante el uso de gnuplot.
7. Implemente un árbol AVL.
  - Implemente recorrido en pre-orden para verificar el balanceo del árbol.
  - Implemente recorrido en in-orden para verificar el balanceo del árbol.
  - Implemente recorrido en pos-orden para verificar el balanceo del árbol.