# **Laboratorio No. 4: Algoritmos voraces**

|  |  |
| --- | --- |
| **Santiago Albisser Cifuentes**  Universidad Eafit  Medellín, Colombia  salbisserc@eafit.edu.co | **Juan Pablo Leal Jaramillo**  Universidad Eafit  Medellín, Colombia  jplealj@eafit.edu.co |

**3) Simulacro de preguntas de sustentación de proyecto**

**3.1** Se usó un arraylist para guardar los sucesores del grafo junto al arreglo de booleanos que permite verificar si se ha visitado un nodo. Luego se revisa si el costo entre el nodo inicial y el sucesor es menor al mínimo, entonces se genera un nuevo costo mínimo y se sigue revisando.

**3.2** Se necesita tener un grafo completo porque de lo contrario van a existir nodos inalcanzables o no visitados entonces el algoritmo voraz no asegurará una óptima solución.

**3.3** Se puede usar para los domicilios visitando los nodos en los que se va a entregar un producto.

**3.4** La única estructura d datos utilizada fueron los arreglos para representar las rutas disponibles. El algoritmo clasifica los arreglos con las rutas en un orden ascendente basado en la duración. La ruta con menor duración en la mañana es emparejada con la ruta de la tarde de mayor duración.

**3.5 O (N log N)**

**3.6 N es el numero de rutas definidas en el problema dado.**

**4) Simulacro de parcial**

* 1. i=j
  2. min>adjacencyMatrix[element][i]

**4.4.1** temp/2

**4.4.2** temp + minimo

**4.4.3**

**4.6.1** i + 1

**4.6.2** res + 1

**4.6.3** i

**4.6.4** 2