

Contexto del Problema:

El gobierno y los alcaldes ha designado a los equipos de la materia de Estructuras y Computación Discretas para que desarrollen un prototipo que permita tener una solución a los colombianos frente a la problemática de los medios masivos de transporte público.

El MIO de Cali vive una crisis que golpea en igual o mayor medida a los sistemas masivos de transporte público del país. Atraviesan fuertes dificultades por la falta de planeación de las rutas más eficientes. Los ciudadanos que hacen uso del transporte público el MIO, se ven obligados a esperar un largo tiempo para dirigirse a su destino, pues los colectivos y las puertas de las paradas se encuentran llenas de personas.

Además, lo que en promedio podría tardar 35 minutos, el sistema masivo de transporte lo hace en 65 minutos, realizando un cálculo, si lo sumamos con dos recorridos mínimos que deben hacer las personas en el día, se está hablando de una hora más en un bus o en un paradero esperando a que llegue la ruta correspondiente.

Un punto importante de esto es que los ciudadanos al observar este fracaso prefieren la compra de autos y motos, haciendo un crecimiento de la malla vial, incrementando la problemática puesto que actualmente los trancones en horas pico pueden hacer que un recorrido de 5 a 10 minutos se transforme en un viaje de 40 a 50 minutos.

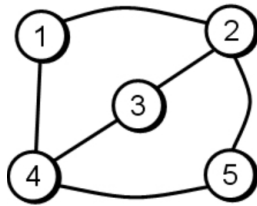
Así que, se requiere crear una solución que permita recorrer todas las estaciones del mío en el menor tiempo posible, y brindarle al usuario cual es la ruta más rápida para llegar a su destino.

Formato TAD:

TAB Grafo

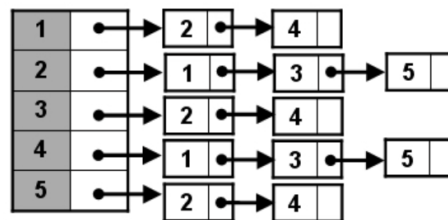
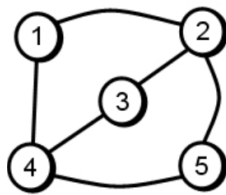
<Objeto abstracto>

- Matriz de adyacencias:



M	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	1	0	1	0	1
5	0	1	0	1	0

- Lista de adyacencias:



inv.: $G=(V, A) \rightarrow V$ es vértice $\wedge A$ es arista \therefore son pares de elementos en V .

Operaciones primitivas:

- crearGrafoVacio : \rightarrow Grafo [ctor]
- añadirArista : Grafo Nodo Nodo \rightarrow [Grafo] [ctor].
- añadirNodo : Grafo Nodo \rightarrow Grafo [ctor] .
- borrarNodo : Grafo Nodo \rightarrow Grafo .
- borrarArista : Grafo Nodo Nodo \rightarrow Grafo .
- esDirecto : Grafo \rightarrow boolean.
- ObtenerCantidadAristas : Grafo \rightarrow int.
- obtenerCantidadNodos : Grafo \rightarrow int.