Carpooling como solución al tráfico

Daniel Felipe Gómez Martínez.
Daniel García García.
César Andrés García Posada.
Medellín, 16-05-19

Estructuras de Datos Diseñada

	Universidad	A	B	C	D
Universidad	0	7	17	27	19
A	7	0	10	15	12
B	17	10	0	10	14
C	27	15	10	0	8
D	19	12	14	8	0

Imagen 1: *Matríz de adyacencia, cada vértice tiene sus vértices adyacentes de una manera ordenada*

Complejidad del Algoritmo

Sub problema	Complejidad	Significado de la variable
Ordenamiento	$O(n^2 \log n)$	Donde n significa la cantidad de Vértices del grafo
Llenar un carro	$O(5n) = 5O(n) = O(n)$	Donde n significa la cantidad de vértices del grafo
Solución Preliminar	$O(n)$	Donde n significa la cantidad de vértices del grafo
Lectura	$O(n^2)$	Donde n significa el número de vértices del grafo
Guardar Archivo	$O(m)$	Donde m significa la cantidad de vehículos
Complejidad Total	**$O(n^2 \log n)$**	

Tabla 1: Complejidad del algoritmo

Explicación del algoritmo

Cantidad de vehículos que se requieren con una constante de tiempo máxima (p)	Conjunto de 5 vértices	Conjunto de 11 vértices	Conjunto de 205 vértices
P = 1.1	-	4 vehículos	61 vehículos
P = 1.2	2 vehículos	4 vehículos	55 vehículos
P = 1.3	-	4 vehículos	51 vehículos
P = 1.7	2 vehículos	-	-

Tabla 2: Resultados Obtenidos

Criterios de Diseño del Algoritmo

Tomando en cuenta que podemos describir puntos específicos en una ciudad como como un grafo y debido a que el problema consta en disminuir la cantidad de vehículos que llegan a un determinado punto – en este caso la universidad –, nos enfocamos en cómo reducir el costo de los trayectos más largos, a través de nuevos caminos, en los cuales se pueden recoger otras personas que posean un costo del trayecto más corto.

Por este motivo decidimos trabajar con LinkedList y ArrayList, debido a que nos proporciona una manera rápida y eficiente a la hora de trabajar con grafos. Si nos fijamos en las tablas de complejidad de cada estructura podemos notar que operaciones como el get() en el ArrayList es de O(1), de igual forma la operación de adicionar (add()) en la LinkedList es de O(1). Adicionalmente, optamos por estas dos estructuras de datos con el fin de ir eliminando cada conjunto de vértices agrupados en un vehículo del ArrayList de la universidad.

Consumo de Tiempo y Memoria

	Constante de tiempo máximo igual a 1.1	Constante de tiempo máximo igual a 1.2	Constante de tiempo máximo igual a 1.3
Caso promedio	50ms	50 ms	48 ms
Peor caso	56 ms	54 ms	52 ms

Tabla 3: *Tiempos de ejecución del algoritmo con diferentes constantes de tiempo máximo para el conjunto de datos de 205 vértices.*

	Constante de tiempo máximo igual a 1.1	Constante de tiempo máximo igual a 1.2	Constante de tiempo máximo igual a 1.3
Mejor caso	1 ms	1ms	0ms
Caso promedio	2ms	2ms	1ms
Peor caso	2 ms	2ms	1 ms

Tabla 4: *Tiempos de ejecución del algoritmo con diferentes constantes de tiempo máximo para el conjunto de datos de 11 vértices.*

	Conjunto de 11 vértices	Conjunto de 205 vértices
Consumo de memoria	2 MG	9 MG

Tabla 5: *Costo en memoria*

Software en funcionamiento

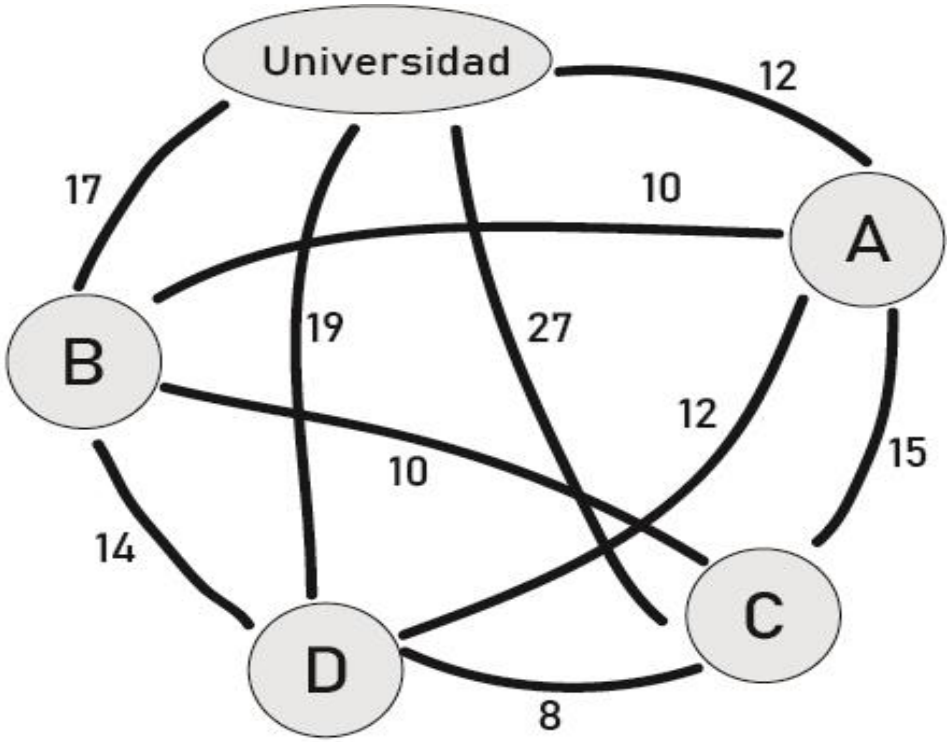

Imagen 2: *Ejemplo de grafo*

Aspectos a destacar:

- ***P = 1.3***
- ***Cantidad máxima de personas en un vehículo = 3***

Universidad	0, Uni	12, A	17, B	19, D	27, C
A	0, A	10, B	12, Uni	12, D	15, C
B	0, B	10, A	10, C	14, D	17, Uni
C	0, C	8, D	10, B	15, A	27, Uni
D	0, D	8, C	15, A	14, B	19, Uni

Imagen 3: *Vértices ordenados de menor a mayor costo*

Universidad	0, Uni	12, A	17, B	19, D	27, C

Imagen 4: *Vértice más alejado de la universidad*

C	0, C	8, D	10, B	15, A	27, Uni

Imagen 5: *Vértice con menor costo de trayecto desde el vértice C*

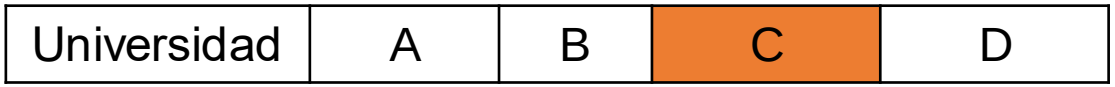

Imagen 6: *Arreglo de vértices visitados*

Imagen 7: *Vértices que conforman un vehículo*

Imagen 8: *Vértices adyacentes al vértice C*

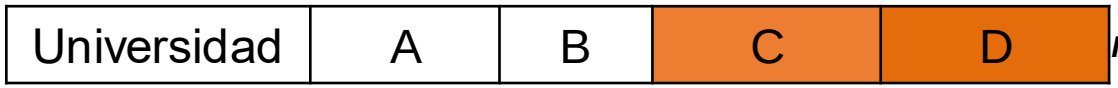

Imagen 9: *Arreglo de vértices visitados*

Imagen 10: *Vértices que conforman un vehículo*

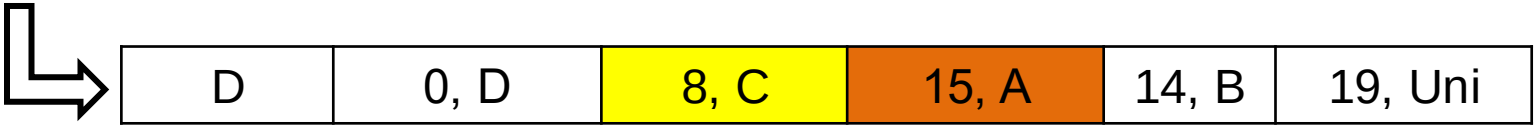

Imagen 11: *Vértices adyacentes al vértice D*

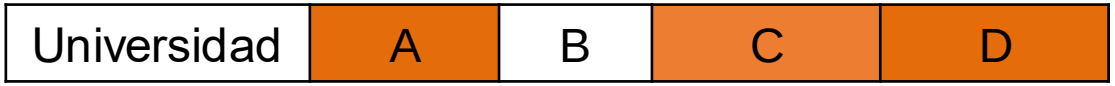

Imagen 12: *Arreglo de vértices visitados*

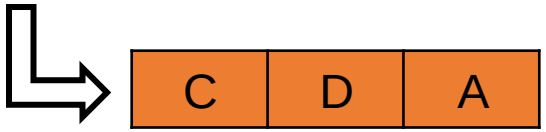

Imagen 13: *Vértices que conforman un vehículo*

Tenemos que en un vehículo van las personas que salen de los vértices C – D – A

Imagen 14: *Vértices que conforman un vehículo*

El vértice que falta por visitar es el B, por ende se usa un vehículo más.

Imagen 15: *Vértices que conforman un vehículo*

En conclusión tenemos dos vehículos

Muchas Gracias