

INFORME DE LECTURA

| | |
|---|---|
| Curso: ST0247 - ESTRUCTURA DATOS Y ALGORÍTMOS 2 | |
| Profesor: Mauricio Toro Bermúdez | |
| Grupo: | 032 |
| Nombre del estudiante: | Daniel Rendon Montaña Juan Daniel Morales Arias Brayam Cristyam Sepulveda Vasquez |
| Código del estudiante: | 201510145010 201519101010 201510077010 |

| | |
|--|--|
| Referencia del artículo: | |
| Nombre de los autores | R.C.T. Lee – S.S. Tseng – R.C. Chang – Y.T. Tsai |
| Nombre del artículo | Introducción al diseño y análisis de algoritmos |
| Libro <u>X</u> Revista <u> </u> Internet <u> </u> Otro <u> </u> | |
| Editorial, Ciudad, Fecha, Núm pág | McGrawHill, Pag 71 - 115 |

Tema del artículo:

El método codicioso (Greedy Algorithms)

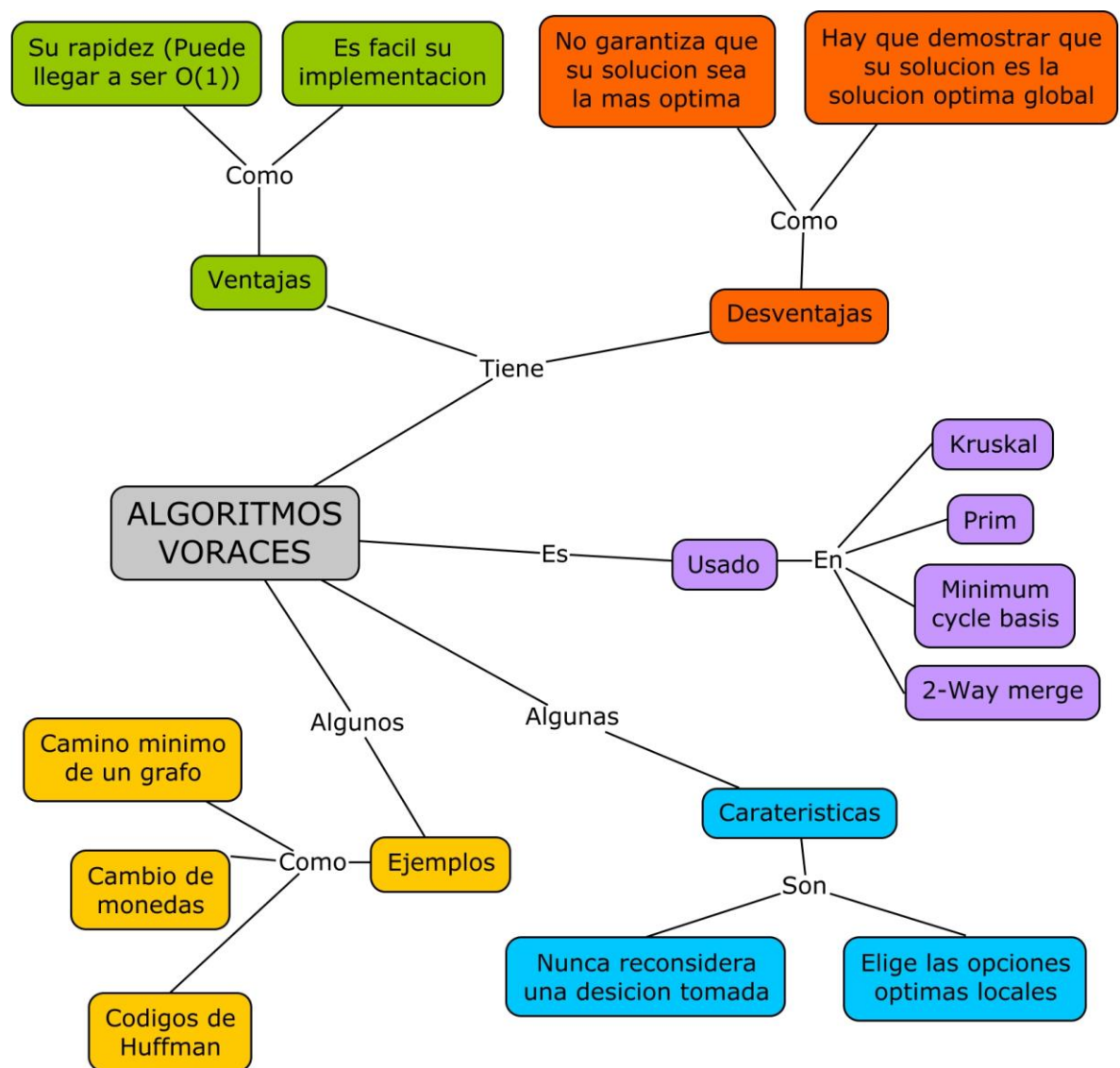
¿Cuánto tiempo (en minutos) le dedicó a la lectura del artículo?:

90 minutos.

Ideas clave (mínimo 5, sustentadas):

1. Los algoritmos voraces sirven para resolver problemas de optimización basados en una secuencia de decisiones. Se basa en elegir la opción optima local, esperando llegar a la solución óptima global
2. Para que los problemas sean solucionables mediante los algoritmos voraces deben cumplir con el Principio de optimalidad de Richard Bellman, el cual dice “dada una secuencia óptima de decisiones, toda subsecuencia de ella es, a su vez, óptima”. Es decir, que estamos seguros que tomando cada optima decisión elemental, podemos llegar a la solución óptima general.
3. El algoritmo de Kruskal es $O(m \log m)$, en el peor de los casos donde m sea n^2 la complejidad será $O(n^2 \log n)$. El algoritmo de Prim tiene complejidad de $O(n^2)$ tanto para el peor de los casos como en caso promedio. Es decir, si el número de aristas del grafo es pequeño, es preferible usar el método de Kruskal, pero en el peor de los casos, el algoritmo de Kruskal se vuelve más lento que el de Prim, por lo cual se recomienda usar el segundo.
4. El algoritmo de Dijkstra nos sirve para encontrar la ruta más corta desde un origen único y lo realiza, en el peor de los casos con n^2 aristas, en una complejidad temporal de $O(n^2)$. Sin embargo, Dijkstra es óptimo en el sentido de los pasos e instrucciones.
5. El algoritmo voraz para mezclar dos listas ordenadas es $O(n \log n)$. Se realiza haciendo un árbol optimo que representa la mezcla de varias listas, donde se mezclan las dos listas de menor elementos sucesivamente hasta llegar a una sola lista.

Representación gráfica de los conceptos más importantes del artículo (Máximo 15 conceptos):



INFORME DE LECTURA

Conclusiones propias – qué aprendió del artículo

- Los algoritmos voraces solo son capaces de resolver algunos problemas de optimización, pero no todos. Aunque la solución óptima local dada por el algoritmo no sea la solución óptima global, esta solución seguirá siendo recomendable porque al menos produce una solución aceptable al problema.
- Los algoritmos voraces no pueden mirar hacia delante. Es decir, “prevenir” lo que viene y usar esto para calcular cual de todas es la mejor solución, en lugar de solo arriesgarse por la mejor en cada movimiento. Esto causa que en muchos casos falle.
- La mezcla de listas es utilizada en el código de Huffman para codificar y comprimir datos, buscando minimizar los costos de transmisión y codificación.

¿Cuánto tiempo (en minutos) le dedicó al análisis del artículo?:

90 minutos.