**Algoritmos Voraces**

1. **Introducción**

Un algoritmo voraz, también conocido como algoritmo greedy, es una estrategia para resolver problemas complejos mediante la toma de decisiones localmente óptimas en cada paso. Aunque no siempre garantizan una solución globalmente óptima, estos algoritmos son populares por su simplicidad, eficiencia y capacidad para encontrar soluciones satisfactorias en muchos casos.

1. **Funcionamiento**

Los algoritmos voraces se basan en un ciclo iterativo que comprende tres etapas:

* **Evaluación:** En cada iteración, se evalúan las opciones disponibles en ese momento.
* **Selección:** Se elige la opción que parece ser la mejor según la estrategia voraz definida.
* **Actualización:** Se actualiza el estado del problema para reflejar la decisión tomada.

El ciclo se repite hasta que se alcanza un estado final que satisface las condiciones de terminación.

1. **Estrategias comunes**

Existen diferentes estrategias voraces que se pueden utilizar en la selección de la mejor opción en cada iteración. Algunas de las más comunes son:

* **El mejor ajuste**: Se elige la opción que mejor se ajusta a un criterio específico, como el menor costo, la mayor ganancia o la menor distancia.
* **El primero en llegar, el primero en ser atendido:** Se elige la opción que ha estado disponible durante más tiempo.
* **El más cercano:** Se elige la opción que está más cerca de un punto de referencia específico.

1. **Análisis de Algoritmos Voraces**

El análisis de un algoritmo voraz implica evaluar su eficiencia y su capacidad para encontrar soluciones óptimas.

* **Eficiencia:** La eficiencia de un algoritmo voraz se mide por su tiempo de ejecución y su complejidad espacial. En general, los algoritmos voraces son bastante eficientes y pueden ejecutarse en tiempo polinomial.
* **Optimalidad:** No todos los algoritmos voraces encuentran soluciones globalmente óptimas. La optimalidad de un algoritmo voraz depende de la naturaleza del problema y de la estrategia voraz utilizada.

1. **Aplicaciones de Algoritmos Voraces**

Los algoritmos voraces se utilizan en una amplia variedad de aplicaciones, incluyendo:

* **Ruteo:** Encontrar la ruta más corta entre dos puntos en un mapa.
* **Asignación:** Asignar tareas a recursos de manera eficiente.
* **Programación:** Seleccionar el mejor conjunto de tareas para ejecutar en un procesador.
* **Optimización:** Encontrar la mejor solución a un problema de optimización, como el problema del vendedor ambulante.

1. **Ejemplos de Algoritmos Voraces**

**Algoritmo de Dijkstra:** Es la ruta más corta entre un vértice y todos los otros vértices de un grafo.

1. **Bibliografía**

* PhinX Lab. (2023, 22 de febrero). ¿Qué es un algoritmo voraz o greedy? Medium. Recuperado de: <https://phinxlab.medium.com/qu%C3%A9-es-un-algoritmo-voraz-o-greedy-6c47ee3b7281>
* Olimpiada Informática. (s. f.). Algoritmia voraz. <https://olimpiada-informatica.org/>

‌