

Açgözlü Algoritmalar Nedir?

Çalışma Kağıdı

Açgözlü algoritmalar, bir problemin çözümünü adım adım oluştururken, her adımda o an için en optimal görünen yerel seçeneği seçen ve bu seçimden geri dönmeyen algoritmik stratejilerdir.

Sorular

1. Açgözlü algoritmalar, bir problemi çözerken hangi stratejiyi izler?
 - A) Her adımda yerel olarak en iyi görünen seçeneği seçer.
 - B) Tüm olası çözümleri dener ve en iyisini seçer.
 - C) Geriye dönük izleme (backtracking) yapar.
 - D) Rastgele seçimler yapar.
2. Aşağıdakilerden hangisi açgözlü algoritmaların bir özelliğidir?
 - A) Optimal çözümü garanti etmesi
 - B) Geriye dönük izleme yapabilmesi
 - C) Yerel optimum seçimleri yapması
 - D) Tüm olasılıkları keşfetmesi
3. Hangi problem türü açgözlü algoritmalarla çözülebilir?
 - A) Sıralama problemleri (örneğin, QuickSort'un pivot seçimi)
 - B) Graf algoritmaları (örneğin, Dijkstra, Kruskal)
 - C) Dinamik programlama problemleri
 - D) Arama algoritmaları
4. En az sayıda madeni para ile belirli bir tutarı ödeme problemi için açgözlü bir yaklaşım nasıldır?
5. Kruskal algoritması ile minimum kapsayan ağacı bulma örneği.
6. Huffman kodlamasında açgözlü yaklaşım nasıl işler?
7. Tanımla: Açgözlü Algoritma Temel Prensipleri Nedir?
8. Tanımla: Açgözlü Algoritmalar Her Zaman Optimal Çözümü Verir mi?
9. Tanımla: Açgözlü Algoritma Örneği Verin.

Cevap Anahtarı

1. A) Her adımda yerel olarak en iyi görünen seçeneği seçer. — Açgözlü algoritmaların temel özelliği, her adımda anlık olarak en avantajlı görünen seçeneği tercih etmesidir.
2. C) Yerel optimum seçimleri yapması — Açgözlü algoritmalar, yerel optimumları seçerek ilerlerler, ancak bu her zaman küresel optimumu garanti etmez.
3. B) Graf algoritmaları (örneğin, Dijkstra, Kruskal) — Dijkstra ve Kruskal gibi algoritmalar, açgözlü stratejiler kullanarak minimum yol veya minimum kapsayan ağaç gibi optimizasyon problemlerini çözer.
4. 1. En büyük değerli madeni paradan başla ve tutarın ne kadarını ödeyebiliyorsan o kadar kullan. 2. Kalan tutar için bir sonraki en büyük değerli madeni parayı kullan. 3. Bu işlemi tutar sıfırlanana kadar tekrarla.
5. 1. Tüm kenarları ağırlıklarına göre sırala. 2. En hafif kenardan başlayarak ilerle. 3. Eğer kenarı eklemek döngü oluşturmuyorsa, ağaca ekle. 4. Tüm düğümler kapsanana kadar devam et.
6. 1. Her karakterin frekansını hesapla. 2. En düşük frekanslı iki karakteri birleştirerek yeni bir düğüm oluştur. 3. Bu işlemi tüm karakterler tek bir kök düğüm altında toplanana kadar tekrarla.
7. Her adımda yerel olarak en iyi seçeneği seçmek.
8. Hayır, her zaman optimal çözümü garanti etmezler; bazı durumlarda suboptimal sonuçlar üretebilirler.
9. Madeni para problemi, Kruskal'ın MST algoritması, Huffman kodlaması.

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.