

Dinamik Programlama Nedir?

Çalışma Kağıdı

Dinamik programlama, bir problemi çözmek için, problemi daha küçük alt problemlere ayırıp, bu alt problemlerin çözümlerini önceden hesaplayıp saklayarak, tekrar eden hesaplamaları önleyen bir yöntemdir.

Sorular

1. Dinamik programlamanın temel avantajı nedir?

- A) Daha az bellek kullanımı
- B) Tekrar eden hesaplamaları önleyerek verimlilik artışı
- C) Daha basit kod yazımı
- D) Her zaman daha hızlı çözüm garantisi

2. Aşağıdakilerden hangisi dinamik programlamanın gerektirdiği özelliklerden biri DEĞİLDİR?

- A) Optimal alt yapı
- B) Üst üste binen alt problemler
- C) Rastgele erişim belleği
- D) Kapsamlı alt problemler

3. Dinamik programlama hangi alanlarda yaygın olarak kullanılır?

- A) Grafik işleme
- B) Doğal dil işleme ve optimizasyon problemleri
- C) Ses sentezi
- D) Veri sıkıştırma algoritmaları

4. Fibonacci serisi hesaplanmasında dinamik programlama nasıl kullanılır?

5. En uzun ortak alt dizi problemini dinamik programlama ile nasıl çözersiniz?

6. Tanımla: Dinamik Programlamanın Temel Fikri Nedir?

7. Tanımla: Hangi Tür Problemler İçin Uygunudur?

8. Tanımla: Memoization Nedir?

Cevap Anahtarı

1. B) Tekrar eden hesaplamaları önleyerek verimlilik artışı — Dinamik programlama, alt problemlerin çözümlerini saklayarak aynı hesaplamaların tekrar yapılmasını engeller ve bu da verimliliği artırır.
2. C) Rastgele erişim belleği — Rastgele erişim belleği (RAM) bilgisayarın genel bir özelliğidir, dinamik programlamanın özel bir gereksinimi değildir. Optimal alt yapı ve üst üste binen alt problemler dinamik programlamanın temelini oluşturur.
3. B) Doğal dil işleme ve optimizasyon problemleri — Dinamik programlama, özellikle en kısa yol problemleri, çanta problemi, en uzun ortak alt dizi gibi optimizasyon problemlerinde ve doğal dil işlemede etkilidir.
4. 1. Temel durumları tanımla: $F(0)=0$, $F(1)=1$. 2. Alt problemler: $F(n) = F(n-1) + F(n-2)$. 3. Çözümleri sakla: Bir dizi veya harita kullanarak hesaplanan her $F(i)$ değerini sakla. 4. Tekrar hesaplamayı önle: Bir $F(i)$ değeri istenirse önce saklananlar arasında ara, yoksa hesapla ve sakla.
5. 1. İki diziyi alın: X ve Y . 2. DP tablosu oluşturun: Boyutları (X 'in uzunluğu+1) x (Y 'nin uzunluğu+1) olan bir tablo (L). 3. Tabloyu doldurun: $L[i][j]$, $X[1..i]$ ve $Y[1..j]$ dizilerinin en uzun ortak alt dizisinin uzunluğunu temsil eder. Eğer $X[i] == Y[j]$ ise $L[i][j] = L[i-1][j-1] + 1$. Değilse $L[i][j] = \max(L[i-1][j], L[i][j-1])$. 4. Sonucu bulun: $L[m][n]$ (m ve n dizilerin uzunluklarıdır) en uzun ortak alt dizinin uzunluğunu verir.
6. Problemi alt problemlere ayırıp çözümleri saklamak.
7. Optimal alt yapı ve üst üste binen alt problemler özelliklerine sahip problemler.
8. Dinamik programlamada alt problemlerin çözümlerini saklama tekniği.

Bounlu

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