

Bellman-Ford Algoritması Nedir?

Çalışma Kağıdı

Bellman-Ford, bir grafikteki tüm kenarları $V-1$ (V düğüm sayısı) kez tarayarak ve her adımda kenar ağırlıklarını güncelleyerek en kısa yolları hesaplar. Eğer V . taramada hala bir güncelleme oluyorsa, bu grafikte negatif döngü olduğunu gösterir.

$$d[v] = \min(d[v], d[u] + w(u, v))$$

$d[v]$ = v düğümüne olan en kısa uzaklık

$d[u]$ = u düğümüne olan en kısa uzaklık

$w(u, v)$ = u 'dan v 'ye giden kenarın ağırlığı

Sorular

1. Bellman-Ford algoritması negatif kenar ağırlıklarına sahip graflarda çalışabilir mi?

- A) Evet
- B) Hayır
- C) Sadece belirli durumlarda
- D) Dijkstra'dan daha hızlıysa

2. Bellman-Ford algoritması bir grafikte negatif döngü olup olmadığını nasıl tespit eder?

- A) İlk geçişte
- B) $V-1$ geçişten sonra
- C) V . geçişte bir güncelleme olursa
- D) Grafiğin yoğunluğuna bakarak

3. Bellman-Ford algoritmasının temel adımı nedir?

- A) En küçük ağırlıklı kenarı seçmek
- B) Tüm kenarları $V-1$ kez gevşetmek
- C) Sadece pozitif ağırlıklı kenarları işlemek
- D) Düğümleri rastgele ziyaret etmek

4. Aşağıdaki grafikte A düğümünden diğer tüm düğümlere en kısa yolu Bellman-Ford ile bulunuz.

5. Bellman-Ford algoritmasının temel amacı nedir?

6. Tanımla: Bellman-Ford hangi tür graflarda çalışır?

7. Tanımla: Dijkstra'dan temel farkı nedir?

8. Tanımla: Algoritmanın çalışma prensibi nedir?

Cevap Anahtarı

1. A) Evet — Bellman-Ford algoritmasının en önemli özelliklerinden biri, negatif kenar ağırlıklarına sahip graflarda da doğru sonuç verebilmesidir.
2. C) V. geçişte bir güncelleme olursa — Algoritma, V-1 geçişten sonra V. geçişte hala bir kenarın gevşetilebildiğini fark ederse, bu grafikte bir negatif döngünün varlığını gösterir.
3. B) Tüm kenarları V-1 kez gevşetmek — Algoritma, grafikteki tüm kenarları V-1 kez tekrarlayarak ve her adımda olası en kısa yolları güncelleyerek çalışır.
4. 1. Başlangıç: A'nın uzaklığı 0, diğer tüm düğümlerin uzaklığı sonsuz olarak ayarlanır. 2. İlk geçiş: A'dan çıkan kenarlar güncellenir. 3. Sonraki geçişler: Grafikteki tüm kenarlar V-1 kez taranarak uzaklıklar güncellenmeye devam eder. 4. Negatif döngü kontrolü: V. geçişte bir güncelleme olup olmadığı kontrol edilir.
5. Bellman-Ford algoritmasının temel amacı, ağırlıklı yönlü bir grafikte, negatif kenar ağırlıkları olsa bile, belirli bir başlangıç düğümünden diğer tüm düğümlere olan en kısa yolları bulmaktır.
6. Ağırlıklı yönlü graflar. Negatif kenar ağırlıklarına izin verir.
7. Dijkstra negatif kenar ağırlıklarında çalışmazken, Bellman-Ford çalışır ve negatif döngüleri tespit edebilir.
8. Kenar ağırlıklarını V-1 kez gevşeterek en kısa yolları bulur.

Bounlu

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