

Dijkstra Algoritması Nedir?

Çalışma Kağıdı

Dijkstra algoritması, ağırlıklı kenarlara sahip bir graf üzerinde, tek bir kaynaktan diğer tüm hedeflere giden en kısa yolları bulmak için kullanılan bir keşif algoritmasıdır. Başlangıç noktasından uzaklıkları artan bir sıra ile ziyaret ederek çalışır.

Sorular

1. Dijkstra algoritması hangi tür graf kenarlarında doğru çalışır?
 - A) Negatif ağırlıklı kenarlar
 - B) Pozitif ağırlıklı kenarlar
 - C) Sıfır ağırlıklı kenarlar
 - D) Tüm kenar türleri
2. Dijkstra algoritması bir yönlendirmeli (greedy) algoritma mıdır?
 - A) Evet
 - B) Hayır
 - C) Duruma göre değişir
 - D) Sadece belirli veri yapıları için
3. Dijkstra algoritmasının temel veri yapısı genellikle hangisidir?
 - A) Yığın (Stack)
 - B) Sıra (Queue)
 - C) Öncelik Kuyruğu (Priority Queue)
 - D) Ağaç (Tree)
4. Dijkstra algoritması ile bir graf üzerinde en kısa yolu nasıl buluruz?
5. Dijkstra algoritmasının temel mantığı nedir?
6. Tanımla: Dijkstra Algoritması Temel Amacı
7. Tanımla: Algoritma Türü
8. Tanımla: Kullanım Alanları

Cevap Anahtarı

1. B) Pozitif ağırlıklı kenarlar — Dijkstra algoritması, negatif kenar ağırlıkları olduğunda yanlış sonuçlar verebilir. Pozitif veya sıfır ağırlıklı kenarlar için tasarlanmıştır.
2. A) Evet — Evet, Dijkstra algoritması her adımda yerel olarak en iyi seçimi yaparak (başlangıç noktasına en yakın düğümü ziyaret ederek) küresel olarak en iyi çözümü bulmaya çalışan bir yönlendirmeli algoritmadır.
3. C) Öncelik Kuyruğu (Priority Queue) — Dijkstra algoritması, ziyaret edilmemiş düğümler arasındaki minimum uzaklığı verimli bir şekilde bulmak için genellikle bir öncelik kuyruğu kullanır.
4. 1. Tüm düğümlerin uzaklığını sonsuz, başlangıç düğümünün uzaklığını 0 olarak ayarla. 2. Ziyaret edilmemiş düğümler kümesini oluştur. 3. Şu anki düğümün komşularının uzaklıklarını güncelle: mevcut düğüme olan uzaklık + kenar ağırlığı. 4. En küçük uzaklığa sahip ziyaret edilmemiş düğümü seç ve ziyaret edildi olarak işaretle. 5. Hedef düğüme ulaşılan veya tüm düğümler ziyaret edilene kadar 3. ve 4. adımları tekrarla.
5. Algoritma, her adımda başlangıç noktasına en yakın olan ve henüz tam olarak işlenmemiş düğümü seçer. Bu düğümün komşularının başlangıç noktasına olan uzaklıklarını, bu yeni düğüm üzerinden geçerek daha kısa bir yol olup olmadığını kontrol ederek günceller. Bu şekilde, her zaman yerel olarak en iyi kararı vererek küresel olarak en kısa yolu bulmaya çalışır.
6. Tek bir kaynaktan, graf üzerindeki tüm diğer noktalara olan en kısa yolları bulmak.
7. Yönlendirmeli (Greedy) Algoritma.
8. Ağ yönlendirme (örn. OSPF), harita uygulamaları, yol planlama.

Bounlu

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