

Dengeli Ağaçlar (AVL, Red-Black) Nedir?

Çalışma Kağıdı

Dengeli ağaçlar, bir ikili arama ağacının yüksekliğini logaritmik olarak sınırlayarak tüm temel işlemlerin (arama, ekleme, silme) ortalama ve en kötü durum karmaşıklığını $O(\log n)$ seviyesinde tutan veri yapılarıdır.

Sorular

- Aşağıdakilerden hangisi dengeli ikili arama ağaçlarının temel amacıdır?
 - Bellek kullanımını en aza indirmek
 - Tüm işlemlerin en kötü durum karmaşıklığını $O(\log n)$ yapmak
 - Düğüm değerlerini sıralı tutmak
 - Ağacın derinliğini artırmak
- Bir AVL ağacında, bir düğümün sol alt ağacının yüksekliği 5 ve sağ alt ağacının yüksekliği 3 ise, bu durum AVL özelliğini ihlal eder mi?
 - Evet, çünkü yükseklik farkı 2'dir.
 - Hayır, çünkü fark 1'den azdır.
 - Evet, çünkü sol alt ağaç daha yüksektir.
 - Hayır, çünkü her iki alt ağaç da dengelidir.
- Red-Black ağaçları, AVL ağaçlarına göre genellikle ekleme ve silme işlemlerinde neden daha hızlı olabilir?
 - Daha sıkı denge kurdukları için
 - Daha az rotasyon gerektirebildikleri için
 - Daha fazla bellek kullandıkları için
 - Her zaman tam dengeli oldukları için
- Bir ikili arama ağacında eleman ekleme işlemi nasıl ağacın dengesini bozabilir?
- AVL ağaçları ile Red-Black ağaçları arasındaki temel fark nedir?
- Tanımla: Dengeli Ağaç Nedir?
- Tanımla: AVL Ağacı Dengesi Kuralı
- Tanımla: Red-Black Ağacı İçin Bir Kural

Cevap Anahtarı

1. B) Tüm işlemlerin en kötü durum karmaşıklığını $O(\log n)$ yapmak — Dengeli ağaçların ana hedefi, ağacın yüksekliğini logaritmik tutarak arama, ekleme ve silme gibi işlemlerin en kötü durum performansını $O(\log n)$ seviyesinde garanti etmektir.
2. A) Evet, çünkü yükseklik farkı 2'dir. — AVL ağaçlarında yükseklik farkı en fazla 1 olmalıdır. Yükseklik farkının 2 olması AVL özelliğinin ihlal edildiği anlamına gelir ve yeniden dengeleme (rotasyon) gerektirir.
3. B) Daha az rotasyon gerektirebildikleri için — Red-Black ağaçları, AVL ağaçlarına göre daha gevşek bir denge mekanizması kullanır. Bu durum, ekleme ve silme işlemleri sırasında genellikle daha az rotasyon yapılmasına olanak tanır, bu da işlemleri potansiyel olarak daha hızlı hale getirir.
4. Eğer elemanlar sıralı bir şekilde eklenirse (örneğin, 1, 2, 3, 4, 5), ağaç bir bağlı listeye dönüşebilir. Bu durumda, en kötü durum arama süresi $O(n)$ olur, çünkü her düğüme sırayla bakmak gerekir. Dengeli ağaçlar, bu tür durumları önlemek için rotasyonlar kullanarak ağacı yeniden dengeler.
5. AVL ağaçları, her düğümün sol ve sağ alt ağaçlarının yükseklikleri arasındaki farkın en fazla 1 olmasını sağlayarak daha katı bir denge kurar. Red-Black ağaçları ise daha gevşek bir denge kullanır; her yolun kökten yaprağa kadar aynı sayıda siyah düğüme sahip olması gibi kurallarla çalışır. Bu gevşeklik, Red-Black ağaçlarında ekleme ve silme işlemlerinin genellikle daha az rotasyon gerektirmesi anlamına gelir.
6. Arama, ekleme ve silme işlemlerinin verimliliğini $O(\log n)$ seviyesinde tutmak için yüksekliği logaritmik olarak sınırlanan ikili arama ağacıdır.
7. Her düğüm için sol alt ağacın yüksekliği ile sağ alt ağacın yüksekliği arasındaki fark en fazla 1 olmalıdır.
8. Her yol, kökten bir yaprağa kadar aynı sayıda siyah düğüme sahip olmalıdır.

Bounlu

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