

Bağlı Listeler Nedir?

Çalışma Kağıdı

Bağlı liste, her biri bir veri ve bir sonraki elemana işaret eden bir işaretçi (veya referans) içeren düğümlerden oluşan bir dizidir.

Sorular

1. Bağlı listede eleman ekleme işlemi, dizilere göre genellikle daha _____.
A) yavaştır
B) hızlıdır
C) karmaşıktır
D) basittir
2. Bağlı listede rastgele erişim (belirli bir indeksteki elemana doğrudan ulaşma) nasıldır?
A) Çok hızlıdır
B) $O(1)$ zaman karmaşıklığındadır
C) $O(n)$ zaman karmaşıklığındadır
D) Mümkün değildir
3. Aşağıdakilerden hangisi bağlı listenin avantajlarından biri DEĞİLDİR?
A) Dinamik boyut
B) Bellek verimliliği (gerektiğinde tahsis)
C) Rastgele erişimde hızlilik
D) Kolay ekleme/silme
4. Basit bir bağlı liste örneği oluşturun.
5. Bağlı listeden belirli bir elemanı nasıl silersiniz?
6. Tanımla: Bağlı Listenin Temel Birimi Nedir?
7. Tanımla: Bağlı Liste Neden Dinamiktir?
8. Tanımla: Tek Yönlü Bağlı Liste Nedir?

Cevap Anahtarı

1. B) hızlıdır — Bağlı listelerde eleman ekleme (özellikle başa veya sona), dizilere göre daha hızlıdır çünkü elemanların kaydırılması gerekmez.
2. C) $O(n)$ zaman karmaşıklığındadır — Bağlı listede rastgele erişim için baştan başlayıp istenen indekse kadar düğümleri tek tek dolaşmak gerekir, bu da $O(n)$ zaman karmaşıklığına yol açar.
3. C) Rastgele erişimde hızlilik — Bağlı listelerin rastgele erişimi yavaştır ($O(n)$), bu avantajlarından biri değildir.
4. 1. Bir 'Düğüm' (Node) sınıfı tanımlayın: veri ve sonraki düğüme işaretçi içerir. 2. Bir 'Bağlı Liste' (LinkedList) sınıfı tanımlayın: ilk düğüme işaretçi (head) içerir. 3. Başa eleman ekleme fonksiyonu yazın: yeni düğüm oluşturun, veri ata, yeni düğümün 'sonraki' işaretçisini mevcut başa ata, başı yeni düğüm yap. 4. Listeyi yazdırma fonksiyonu yazın: baştan başlayarak her düğümün verisini yazdır ve 'sonraki' işaretçisiyle devam et.
5. 1. Silinecek elemanın değerini bilin. 2. Listenin başından başlayarak ilerleyin. 3. Mevcut düğümün 'sonraki' düğümünün verisini kontrol edin. 4. Eğer 'sonraki' düğüm silinecek elemansa, mevcut düğümün 'sonraki' işaretçisini, silinecek düğümün 'sonraki' işaretçisine bağlayın. 5. Eğer baş eleman silinecekse, listenin başını (head) bir sonraki elemana güncelleyin.
6. Düğüm (Node). Her düğüm veri ve bir sonraki düğüme işaretçi içerir.
7. Bellek tahsisi çalışma zamanında yapılabilir ve liste boyutu ihtiyaç duyuldukça değişebilir.
8. Her düğümün yalnızca bir sonraki düğüme işaret ettiği bağlı liste türüdür.

Bounlu

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