

Bağlı Listeler Nedir?

Çalışma Kağıdı

Bağlı Liste, her bir düğümün veriyi ve bir sonraki düğüme işaret eden bir işaretçiyi (pointer) içerdiği, doğrusal bir veri yapısıdır.

Sorular

1. Bağlı listede rastgele erişim süresi nedir?
A) Sabit zaman ($O(1)$)
B) Doğrusal zaman ($O(n)$)
C) Logaritmik zaman ($O(\log n)$)
D) Karesel zaman ($O(n^2)$)
2. Dizilere (Array) göre bağlı listelerin en büyük avantajı nedir?
A) Daha hızlı arama
B) Sabit bellek kullanımı
C) Dinamik boyutlanma ve verimli ekleme/silme
D) Daha kolay indeksleme
3. Bağlı listenin sonuna eleman eklemek için kaç işlem gerekir?
A) $O(1)$
B) $O(n)$
C) $O(\log n)$
D) $O(n^2)$
4. Basit bir bağlı liste oluşturma ve eleman ekleme örneği nedir?
5. Bağlı listeden eleman silme işlemi nasıl yapılır?
6. Bağlı listelerin kullanım alanları nelerdir?
7. Tanımla: Bağlı Liste Nedir?
8. Tanımla: Düğüm (Node) Nedir?
9. Tanımla: Head (Baş) Nedir?

Cevap Anahtarı

1. B) Doğrusal zaman ($O(n)$) — Bağlı listede bir elemana erişmek için listenin başından başlayıp sırayla ilerlemek gerekir, bu da eleman sayısı ile orantılı bir süre ($O(n)$) alır.
2. C) Dinamik boyutlanma ve verimli ekleme/silme — Dizilerin boyutu sabittir ve ortasına eleman eklemek/silmek maliyetlidir. Bağlı listeler ise dinamik olarak büyüyüp küçülebilir ve ekleme/silme işlemleri daha verimlidir.
3. B) $O(n)$ — Listenin sonuna eleman eklemek için listenin sonuna kadar ilerlemek gerekir, bu da eleman sayısına bağlıdır ($O(n)$). Listenin son düğümünü ayrıca tutan bir yapı (tail pointer) ile bu $O(1)$ yapılabilir.
4. Bir 'Düğüm' sınıfı tanımlayın: Her düğüm bir 'veri' alanı ve bir 'sonraki' işaretçisi içermelidir. Bir 'BağlıListe' sınıfı oluşturun: Bu sınıf, listenin başını (head) tutan bir işaretçi içermelidir. Başlangıçta 'head' null olarak ayarlayın. Listenin sonuna eleman eklemek için: Yeni bir düğüm oluşturun. Eğer liste boşsa, yeni düğümü 'head' yapın. Değilse, listenin sonuna kadar ilerleyin ve son düğümün 'sonraki' işaretçisini yeni düğümüne ayarlayın.
5. Silinecek düğümün değerini veya konumunu belirleyin. Listenin başından başlayarak ilerleyin. Önceki düğümü takip edin. Silinecek düğümü bulduğunuzda, önceki düğümün 'sonraki' işaretçisini, silinecek düğümün 'sonraki' işaretçisine ayarlayın. Eğer silinecek düğüm listenin başındaysa, 'head' işaretçisini listenin ikinci elemanına güncelleyin.
6. Uygulamalarda dinamik bellek yönetimi gerektiren yerlerde (örn. yığınlar, kuyruklar). İşletim sistemlerinde süreç yönetimi. Tarayıcılarda 'geri' ve 'ileri' tuşlarının işleyişi (çift bağlı liste ile). Veritabanlarında indeksleme yapıları.
7. Elemanların (düğümün) veriyi ve bir sonraki elemana işaretçiyi tuttuğu doğrusal veri yapısı.
8. Bağlı listenin temel yapı taşıdır; veri ve bir sonraki düğümüne işaretçi içerir.
9. Bağlı listenin ilk düğümünü gösteren işaretçidir.

Bounlu

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