Neden Bağlantılı Liste? (array ile karşılaştırma)

Diziler, benzer türlerdeki doğrusal verileri depolamak için kullanılabilir, ancak dizilerin aşağıdaki sınırlamaları vardır.

1) Dizilerin boyutu sabittir, eleman sayısının sınırını önceden bilmeliyiz.

2) Bir dizi elemana yeni bir eleman eklemek pahalıdır, çünkü listeye yeni eleman eklenmesi için mevcut elemanların kaydırılması gerekir.

Örneğin, bir sistemde, bir dizi [] sıralı bir liste tutarsak.

dizi [] = [22, 4, 2, 8, 12].

Ve eğer yeni bir eleman olarak 2 den sonrasına eklemek istiyorsak, sıralı düzeni korumak için, tüm 2'den sonra olan elemanları taşımamız gerekir (2 hariç). Silme, bazı özel teknikler kullanılmadıkça dizilerde de pahalıdır.

Dizilere göre avantajları

1.Dinamik olması 2. Silme kolaylığı

Dezavantajlar

1. Rastgele erişime izin verilmez. İlk düğümden başlayarak öğelere sırayla erişmemiz gerekiyor. Bu nedenle, varsayılan uygulamasıyla bağlantılı listelerle ikili arama yapamayız.
2. Listenin her öğesi için işaretçi tutuyoruz. Bunun için fazladan bellek alanı gereklidir.
3. Önbellek dostu değildir. Dizi elemanları bitişik konumlar olduğu için, bağlantılı listelerde bulunmayan referans yeri vardır.

Temsil: Bağlantılı bir liste, bağlantılı listenin ilk düğümüne bir gösterici ile temsil edilir. İlk düğüme kafa denir. Bağlantılı liste boşsa, başın değeri NULL olur. Listedeki her düğüm en az iki bölümden oluşur: 1) veriler 2) Sonraki düğüme işaretçi (veya referans) C'de, yapıları kullanarak bir düğümü temsil edebiliriz. Aşağıda, tamsayı verileriyle bağlantılı bir liste düğümü örneği bulunmaktadır. Java veya C #'da LinkedList bir sınıf olarak ve bir Node ayrı bir sınıf olarak temsil edilebilir. LinkedList sınıfı, Node sınıf türünün bir başvurusunu içerir.