ALGORİTMA ANALİZİ ÖDEV RAPORU

Lineer Prob Hashing

Grup 1 – Mine Elif Karslıgil

MÜDAFER KAYMAK

20011093

Mudafer.kaymak@std.yildiz.edu.tr

Problem Tanımı

Ödevin amacı, bir sistemde bulunan kullanıcı isimlerini etkili bir şekilde saklamak ve bu isimler üzerinde ekleme, arama ve silme işlemlerini gerçekleştirmek için bir hash tablosu oluşturmaktır. Hash tablosu, veri erişimi için hızlı ve verimli bir yöntem sağlamak amacıyla kullanılacaktır.

Problemin Çözümü

Bu ödev kapsamında, kullanıcı isimlerini etkili bir şekilde saklamak üzere hash tablosu oluşturulmaktadır. Hash tablosu, open address ve double hashing yöntemleri kullanılarak implemente edilmekte olup, çakışma problemlerini çözmek adına etkili bir çözüm sunmaktadır. Hash tablosunun oluşturulması sırasında bölme yöntemiyle hash fonksiyonu hesaplamaları gerçekleştirilmekte ve Horner Kuralı kullanılarak kullanıcı isimlerinin sayısal karşılıkları belirlenmektedir. Ayrıca, tablo uzunluğu ve load factor kullanıcıdan alınarak, tablonun boyutu ve performansı optimize edilmektedir.

Karşılaşılan Zorluklar

Silme işlemlerinde deleted alanı kullanılarak gerçekleştirilen bir yaklaşım benimsendi. Ancak, bu durumun kullanıcı arayüzünde nasıl yönetileceği ve kullanıcılara doğru bilgi aktarımı sağlanması konusunda başlangıçta belirsizlikler yaşandı. Silinen bir kullanıcı adının ardından yapılacak aramalarda doğru mesajın verilmesi ve aynı isimli yeni bir kullanıcının eklenmesi durumunda nasıl bir işlem yapılacağına dair detaylar üzerinde çalışmalar yapıldı. İlgili literatür incelenerek çeşitli test senaryoları oluşturuldu ve bu senaryolarda performans analizleri gerçekleştirildi. Böylece, karşılaşılan zorlukların üstesinden gelinerek etkili bir hash tablosu çözümü elde edildi.

Karmaşıklık Analizi

Ekleme İşlemi (Insert): Ekleme işlemi, hash tablosuna yeni bir kullanıcı eklerken kullanılan hash fonksiyonlarının ve çakışma çözüm stratejilerinin etkinliğini değerlendirmek açısından önemlidir. Hash fonksiyonlarının karmaşıklığı O(1) düzeyinde olmalıdır. Double hashing ve bölme yöntemi kullanıldığı için, ekleme işlemi ortalama durumda O(1) karmaşıklığına sahiptir.

Arama İşlemi (Search): Arama işlemi, kullanıcının hash tablosunda bulunan bir ismi sorgularken kullanılan hash fonksiyonlarının ve çakışma çözüm stratejilerinin etkinliğini değerlendirir. Arama işlemi de ortalama durumda O(1) karmaşıklığına sahiptir.

Silme İşlemi (Delete): Silme işlemi, kullanıcı isminin tablodan silinmesini ve deleted alanının güncellenmesini içerir. Silme işlemi de ortalama durumda O(1) karmaşıklığına sahiptir. Ancak, silme işlemlerinin ardından tablonun boyutunun optimize edilmesi adına güncellenmiş bir tablo oluşturulması, işlemleri etkilemektedir.

Bu analizler sonucu karmaşıklık Θ(1) çıkacaktır.

Ekran görüntüleri