

## 2021-2022 Güz Yarıyılı Algoritma Analizi Final Projesi

Ders Yürütücüleri  
**Prof. Dr. M. Elif KARSLIGİL**  
**Doç. Dr. M. Amaç GÜVENSAN**

### Konu: Hashing

**Problem:** Bu ödevde erişimi hızlandırmak için hashing yönteminden faydalanan bir LRU Cache (En Son Kullanılan Ön Bellek) tasarlanmanız istenmektedir.

**1. LRU İşlemleri:** LRU Cache, hızlı erişim için son kullanılan N adet veriyi, en son kullanılabilecek şekilde linkli listede saklayan bir veri yapısıdır. Bu ödevde cache üzerinde kişilerin kimlik-no, ad soyad, doğum tarihi ve adres bilgileri saklanacaktır. Bir kişinin bilgilerinin cache üzerinde olup olmadığını tespit etmek için linkli liste üzerinde arama yapmak yerine, **kimlik-no bilgisine göre** hash tablosuna yerleştirilecektir. Kimlik-no ve linkli listede kaçınıcı sırada oldukları hash tablosunda tutularak arama işleminin hızlandırılması sağlanacaktır. Erişilmek istenen kimlik-no verildiğinde aşağıdaki işlemler yapılmalıdır:

1. Kimlik-no bilgisini hash tablosunda arayınız.
2. Kimlik-no hash tablosunda **varsa**
  - a. linkli listenin kaçınıcı sırasında ise buna göre linkli liste üzerinde kişinin bilgilerine erişerek ekrana yazdırınız.
  - b. bu kişinin bilgilerini linkli listenin en başına alıp, hash tablosunda da yer bilgisini güncelleyiniz.
3. Kimlik-no ismi hash tablosunda **yoksa**
  - a. Cache'deki eleman sayısı N'den az ise yeni elemanı linkli listenin başına ve hash tablosuna ekleyiniz.
  - b. Cache dolu ise, linkli listenin en sonundaki kişi bilgilerini hem linkli listeden, hem de hash tablosundan siliniz. Yeni elemanı linkli listenin başına ve hash tablosuna ekleyiniz.

**2. Hash Tablosu İşlemleri:** Hash tablosu üzerinde aşağıdaki işlemleri yapınız.

1. Tabloya yerleştirilecek kimlik-no bilgisinin sayılardan oluşan uzun bir string olduğunu varsayarak bu string'in sayı karşılığını Horner Kuralı kullanılarak hesaplayınız.
2. Hash tablosunu oluştururken openaddress, çakışma(collusion) problemini çözmek için double hashing yöntemini kullanınız. Hash fonksiyonlarını aşağıdaki bağıntıları kullanarak belirleyiniz:
 
$$h(\text{key}, i) = (h1(\text{key}) + i * h2(\text{key})) \% m$$

$$h1(\text{key}) = \text{key} \% m$$

$$h2(\text{key}) = 1 + (\text{key} \% m2)$$
 h2 fonksiyonundaki m2 değerini  $m2 = m - 1$  olarak alınız.
3. Tablo uzunluğunu gösteren m değerinin belirlenmesi için aşağıdaki bağıntıyı kullanınız:
 
$$\text{TabloUzunluğu} = \text{EnküçükAsalSayı} \geq \text{CacheUzunluğu} / \text{LoadFactor}$$
 LoadFactor'u 0.60 alınız. Tablo uzunluğunu dışardan giriş olarak alınız.
4. Hash tablosunda her eleman için kimlik-no, bu kişinin linkli listede sırası ve kişiye ait bilgilerin cache'den silinip silinmediği bilgisini saklayınız. Bir eleman cache'den silindi ise hash tablosunda bu adrese gelen yeni bir elemanı yerleştirebilirsiniz.

**Ödev Teslimi:**

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **09.01.2022 saat 23:59'a** kadar online.yildiz.edu.tr üzerinden **HW4\_OgrenciNumarasi.rar** dosyasını yükleyiniz.

- Ödev kontrolleriniz **uzaktan** yapılacaktır.
- Detaylı bilgi için **Arş. Grv. Fuat ÖGME'nin** [avesis.yildiz.edu.tr](http://avesis.yildiz.edu.tr) adresindeki sayfasını takip ediniz.

**Teslim Edilecekler:**

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız. **Yöntem** bölümünde problemi ve çözüm için önerdiğiniz yöntemi kısaca anlatınız.
2. Algoritmanızın cache belleğe eleman ekleme, arama ve silme işlemleri için zaman ve yer karmaşıklıklarını hesaplayınız.
3. **Uygulama** bölümünde yapmanız gerekenler için Arş. Grv. Fuat Ögme'nin sayfasını takip ediniz.
4. Algoritmanızın **C** dilinde programını hazırlayarak dokümana da ekleyiniz.
5. **Teslim Edilecekler**
  - a. HW#\_OgrenciNumarasi.rar (Örn: HW4\_15011001.rar)
    - i. OgrenciNumarasi.pdf (Örn: 15011001.pdf)
    - ii. OgrenciNumarasi.c (Örn: 15011001.c)

**Değerlendirme:**

**Bilgisayar bilimleri anabilim dalı araştırma görevlilerinin [avesis](http://avesis.yildiz.edu.tr) sayfalarında yer alan Ödev Teslim Kuralları başlığındaki kurallara uyulması gerekmektedir.**

***Algoritma Tasarımı ve Programın Çalışması: (%80)***

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Program hatasız çalışmalıdır.
4. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

***Rapor Dokümantasyonu: (%20)***

1. Raporun kapak sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
3. Değişken isimleri anlamlı olmalıdır.
4. Her fonksiyonun yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
5. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
6. Gereksiz kod tekrarı olmamalıdır.
7. Kaynak kodun formatı düzgün, okunabilir ve takip edilebilir olmalıdır.