

**LEAST RECENTLY USED ALGORITHM
IMPLEMENTATION**

09.01.2022

**ALGORİTMA ANALİZİ – BLM3021
4. ÖDEV RAPORU**

18011047 – BERKAY KOÇ

YÖNTEM:

Hafızanın verimli kullanılabilmesi ve gereksiz işlemten oldukça kaçınmak için oluşturulan bu çözümde cache yapısı ile kişilerin bilgileri saklanmış, cache üzerinde arama yapma sayısını kısıtlamak için ise hash tablosunda bu değerler kişinin kimlik numarası ile yine kimlik numaralarının Horner metodu kullanılarak oluşturulan değerinin bulunduğu indiste temsil edilmiştir. Hash tablosunda bir yerin üç durumu da(dolu, boş, silinmiş) incelenmiştir. Arama sayısını kısıtlamak için ise cache yapısındaki elemanın sırası da hash tablosunda saklanmıştır. Oluşturulan struct yapıları aşağıdaki gibidir.

```
struct hash{
    char id[10]; //tc kimlik numarasını saklayacak olan karakter dizisi
    int order; //kişinin cache bellekte hangi sırada olduğunu ifade eden integer değer
    int flag; //hash tablosundaki gözün dolu, boş veya silinmiş olma durumunu ifade eden değer
};
```

```
struct personList{
    char id[10]; //tc kimlik numarasını saklayacak olan karakter dizisi
    int birthDate; //doğum tarihini saklayacak olan integer
    char fullName[20]; //kişinin isim-soyismini saklayacak olan karakter dizisi
    char city[10]; //kişinin şehrini saklayacak olan karakter dizisi
    struct personList* next; //Linki listenin sonraki elemanını ifade eden pointer
};
```

Fonksiyonları açıklamak gerekirse;

- `int calculatePrimeValue(int n)`: Cache boyutundan büyük en küçük asal sayıyı hesaplar. Brute force yerine zaman karmaşıklığı bakımından daha optimize bir yöntem kullanılmıştır. İstenen sayının iki katına kadar olan bir dizi açılır(asal sayı bu aralıkta olmak zorundadır) ve ilk yarıdaki sayıların katları işaretlenerek asal olmadıkları belirlenir. İşaretlenmeyen sayılar $0-2*n$ arasındaki tüm asal sayılar bulunmuş olur. Bundan sonra verilen sayıdan büyük en küçük asal sayı bulunur.
- `int keyValue(char* id)`: TC kimlik numarası karakter dizisini alarak horner metodu ile key değerini hesaplar.
- `int calculateHashIndex(int i, int key, int tableSize)`: Verilen indis değerini kullanarak double hashing ile hash tablosundaki gözü belirler.
- `void insertToHashTable(struct hash *hashTable, char id[10], int tableSize)`: TC kimlik numarası verilen kişiyi hash tablosuna ekler.
- `struct personList *deleteFromCache(struct personList* persons, struct hash* hashTable, int tableSize)`: Cache'in dolması durumunda listenin sonuna kadar giderek son elemanı siler ve free eder.
- `struct personList *insertToCache(struct personList* persons, char id[10], int bDate, char firstName[10], char lastName[10], char city[10])`: Verilen parametreleri kullanarak bir liste oluşturur ve parametre olarak gönderilen listeyi bunun next'ine ekler. Main içerisinde bu fonksiyonun dönüş değeri yeni cache olarak atanır ve kullanılır.
- `int search(char id[10], struct hash* hashTable, int tableSize)`: Hash tablosunda horner metodu ile TC kimlik numarası hesaplanmış kişiyi aratır. bulunamazsa -1, bulunursa index numarası döndürülür.
- `struct personList *updateCache(struct personList* persons, int order)`: Cache içerisindeki eleman tekrar girildiyse cache listesinin güncellenmesi, o ismin en başa gelmesi gerekir. Bu işlemler updateCache fonksiyonu içerisinde gerçekleşir. Geriye bir struct personList pointeri döndürülür ve bu liste yeni cache olarak mainde kullanılır.

Zaman Karmaşıklığı:

Eleman ekleme işlemi için zaman karmaşıklığı başa ekleme işlemi olduğundan, $O(1)$ 'dir.

Eleman silme işlemi için zaman karmaşıklığı bellek 1 kez dolaşıldığı için ve hash tablosundaki değişikliklerden dolayı $O(n)$ 'dir.

Eleman arama işlemi için zaman karmaşıklığı hash tablosunda bu işlem yapıldığı için $O(1)$ 'dir

Yer Karmaşıklığı:

Eleman ekleme işlemi için yer karmaşıklığı linkli listeye ve hash tablosuna eleman eklendiğinden $O(1)$ 'dir

Eleman silme işlemi için yer karmaşıklığı linkli listeye ve hash tablosundan eleman silindiğinden $O(1)$ 'dir

Eleman arama işlemi için yer karmaşıklığı yoktur.

UYGULAMA:

```
Please enter the cache size: 4
Table size will be: 7
Person information: 12345, SABAN DEMIRHAN, 6553044, istanbul
12345, inserted to the index: 1
Elements in the cache:
12345, SABAN DEMIRHAN, 1993, istanbul
Elements in the hash table:
TCKN, Order, flag
0--empty, 0, -1
1--12345, 1, 1
2--empty, 0, -1
3--empty, 0, -1
4--empty, 0, -1
5--empty, 0, -1
6--empty, 0, -1
```

Cache boyutu olarak 4 girilmiştir. Hash tablosunun uzunluğu 7 olarak hesaplanmış, gelen ilk kişi cache belleğe ve hash tablosuna eklenmiştir. Insert edildiği yer görüldüğü gibi 1. İndistir.

```
-----
Person information: 32145, TEYFIK ALBEN, 6553044, ankara
32145, inserted to the index: 6
Elements in the cache:
32145, TEYFIK ALBEN, 1984, ankara
12345, SABAN DEMIRHAN, 1993, istanbul
Elements in the hash table:
TCKN, Order, flag
0--empty, 0, -1
1--12345, 2, 1
2--empty, 0, -1
3--empty, 0, -1
4--empty, 0, -1
5--empty, 0, -1
6--32145, 1, 1
```

Dosyadan 2. kişi okunmuştur. Hash tablosunda bulunamadığından cache belleğe ve hash tablosuna eklenmiştir. Elemanların sıraları baştan düzenlenmiş, flagler değiştirilmiştir.

```

-----
Person information: 43213, AYSU SATIROGLU, 6553044, istanbul

43213, inserted to the index: 0
Elements in the cache:
43213, AYSU SATIROGLU, 1991, istanbul
32145, TEYFIK ALBEN, 1984, ankara
12345, SABAN DEMIRHAN, 1993, istanbul

Elements in the hash table:
TCKN, Order, flag
0--43213, 1, 1
1--12345, 3, 1
2--empty, 0, -1
3--empty, 0, -1
4--empty, 0, -1
5--empty, 0, -1
6--32145, 2, 1

```

3. kişi dosyadan okunmuş, hash tablosunda bulunamadığından cache belleğe ve hash tablosuna eklenmiştir. Elemanların sıraları baştan düzenlenmiş, flagler değiştirilmiştir.

```

-----
Person information: 45543, EMIRCAN GOKMEN, 6553044, ankara

45543, inserted to the index: 2
Elements in the cache:
45543, EMIRCAN GOKMEN, 2003, ankara
43213, AYSU SATIROGLU, 1991, istanbul
32145, TEYFIK ALBEN, 1984, ankara
12345, SABAN DEMIRHAN, 1993, istanbul

Elements in the hash table:
TCKN, Order, flag
0--43213, 2, 1
1--12345, 4, 1
2--45543, 1, 1
3--empty, 0, -1
4--empty, 0, -1
5--empty, 0, -1
6--32145, 3, 1

```

4. kişi dosyadan okunmuş, hash tablosunda bulunamadığından cache belleğe ve hash tablosuna eklenmiştir. Elemanların sıraları baştan düzenlenmiş, flagler değiştirilmiştir.

```

-----
Person information: 43321, ZUBEYDE HARMANBASI, 6553044, izmir

inserted to the index: 4
Elements in the cache:
43321, ZUBEYDE HARMANBASI, 2001, izmir
45543, EMIRCAN GOKMEN, 2003, ankara
43213, AYSU SATIROGLU, 1991, istanbul
32145, TEYFIK ALBEN, 1984, ankara

Elements in the hash table:
TCKN, Order, flag
0--43213, 3, 1
1--deleted, 0, 0
2--45543, 2, 1
3--empty, 0, -1
4--43321, 1, 1
5--empty, 0, -1
6--32145, 4, 1

```

5. kişi listede bulunamadığından ve cache dolu olduğundan son eleman hashten ve cacheden çıkarılır ve yeni gelen kişi başa eklenir. Hash tablosunda sıra düzenlenir ve flag değerleri güncellenir.

```

-----
Person information: 54213, AYSEL OZBEK, 6553044, trabzon

inserted to the index: 5
Elements in the cache:
54213, AYSEL OZBEK, 2000, trabzon
43321, ZUBEYDE HARMANBASI, 2001, izmir
45543, EMIRCAN GOKMEN, 2003, ankara
43213, AYSU SATIROGLU, 1991, istanbul

Elements in the hash table:
TCKN, Order, flag
0--43213, 4, 1
1--deleted, 0, 0
2--45543, 3, 1
3--empty, 0, -1
4--43321, 2, 1
5--54213, 1, 1
6--deleted, 0, 0

```

6. kişi hash tablosunda bulunamadığından başa eklenir ve cache dolu olduğundan son eleman çıkartılır. Hash tablosunda elemanların sıraları güncellenir. Yeni eklenen kişinin flag değeri ve çıkan kişinin flag değeri değiştirilir.

```

-----
Person information: 43321, ZUBEYDE HARMANBASI, 6553044, izmir

Elements in the cache:
43321, ZUBEYDE HARMANBASI, 2001, izmir
54213, AYSEL OZBEK, 2000, trabzon
45543, EMIRCAN GOKMEN, 2003, ankara
43213, AYSU SATIROGLU, 1991, istanbul

Elements in the hash table:
TCKN, Order, flag
0--43213, 4, 1
1--deleted, 0, 0
2--45543, 3, 1
3--empty, 0, -1
4--43321, 1, 1
5--54213, 2, 1
6--deleted, 0, 0

```

Yeni gelen 7. kişi hash tablosunu kontrol neticesinde listede olduğundan cache belleğin en başına getirilir. Hash tablosunda yalnızca elemanların sıraları güncellenir.

```

-----
Person information: 33445, ACELYA SENLIK, 6553044, adana

33445, inserted to the index: 1
Elements in the cache:
33445, ACELYA SENLIK, 1990, adana
43321, ZUBEYDE HARMANBASI, 2001, izmir
54213, AYSEL OZBEK, 2000, trabzon
45543, EMIRCAN GOKMEN, 2003, ankara

Elements in the hash table:
TCKN, Order, flag
0--deleted, 0, 0
1--33445, 1, 1
2--45543, 4, 1
3--empty, 0, -1
4--43321, 2, 1
5--54213, 3, 1
6--deleted, 0, 0

```

8. eleman hash tablosundaki kontrolden sonra bulunamaz ve belleğin başına eklenir. Cache belleğin sonundaki eleman hash tablosundan ve bellekten silinir.

```

-----
Person information: 12345, SABAN DEMIRHAN, 6553044, istanbul

inserted to the index: 3
Elements in the cache:
12345, SABAN DEMIRHAN, 1993, istanbul
33445, ACELYA SENLIK, 1990, adana
43321, ZUBEYDE HARMANBASI, 2001, izmir
54213, AYSEL OZBEK, 2000, trabzon

Elements in the hash table:
TCKN, Order, flag
0--deleted, 0, 0
1--33445, 2, 1
2--deleted, 0, 0
3--12345, 1, 1
4--43321, 3, 1
5--54213, 4, 1
6--deleted, 0, 0

```

Dosyadan okunan yeni kişi hash tablosunda bulunamaz ve hem belleğin başına hem hash tablosuna eklenir. Sondaki eleman hem hash tablosundan hem cache bellekten silinir.

```

-----
Person information: 33445, ACELYA SENLIK, 6553044, adana

Elements in the cache:
33445, ACELYA SENLIK, 1990, adana
12345, SABAN DEMIRHAN, 1993, istanbul
43321, ZUBEYDE HARMANBASI, 2001, izmir
54213, AYSEL OZBEK, 2000, trabzon

Elements in the hash table:
TCKN, Order, flag
0--deleted, 0, 0
1--33445, 1, 1
2--deleted, 0, 0
3--12345, 2, 1
4--43321, 3, 1
5--54213, 4, 1
6--deleted, 0, 0

```

Hash tablosunda aranan eleman bulunur. Listenin başına getirilir. Order sıraları ve flag değerleri hash tablosunda güncellenir.