



BLM2512 Veri Yapıları ve Algoritmalar

**Cache Buffer (Önbellek Tamponu) Tasarımı**

**Öğrenci Adı Soyadı :** *Berat Can KARAKAŞ*

**Öğrenci No :** *21011102*

**Dersin Eğitmeni :** *Mine Elif KARSLIGİL*

**Video Linki :** <https://www.youtube.com/watch?v=HMSZpFNO2AA>

# 1 - Problem Tanımı - - -

Bu algoritma yapısında ikili linkli liste yapısı kullanılarak dosyadan okunulan , ve içerisinde ulaşmak istenen adresleri tutulduğu bir veri yapısı oluşturulması istenmiştir . Bu problemde aynı adrese ulaşmak istenmesi durumunda bir sayaç yardımı ile bu erişim isteklerinin tutulması istenmiştir . Bir adrese kullanıcıdan alınan bir eşik değerin üzerinde erişim isteğinde bulunması durumunda bu erişim istenen adresin bulunduğu düğüm ikili linkli listenin ilk elemanı olarak tutturulması , ayrıca Linki Listenin yine kullanıcıdan alınan değer kadar boyutu bulunmaktadır , bu boyutun aşıldığı durumlarda da en sonda bulunan düğümün listeden silinmesi istenmiştir .

# 2 - Problem Çözümü - - -

Kullanıcıdan ilk önce maksimum düğüm sayısı ve Eşik değerleri sırasıyla istenmiştir . Daha sonra dosyadan okunacağı için kullanıcıdan istediği dosyanın adı istenmiştir .

Kullanıcıdan alınan dosya ile o dosyaya erişim sağlanarak değerler satır satır okunmaya başlanmıştır . Okunan bilgi listenin içinde var mı diye kontrol edildikten sonra yok ise yeni bir düğüm oluşturularak dosyadan okunan değerler aktarılmıştır ve sayaç değeri 1 olarak düzenlenmiştir . Eğer listenin içinde var olan bir bilgi olursa , sayaç 1 artırılmıştır . Dosyanın içindeki veriler tamamen böyle okunmuştur . Bu işlem esnasında da ayrıca başta kullanıcıdan istenen eşik değeri ve maksimum düğüm değeri için kontroller yapılmıştır. Yeni bir düğüm eklenmesi durumunda maksimum düğüm sayısının aşıp aşılmadığı kontrol edilmiş ve eğer aşılmış ise son düğüm silinmiştir. Var olan bir düğüm üzerinde sayaç arttırma işlemi yapılmış ise burada da sayaç istenen eşik değerini geçmiş mi diye kontroller sağlanmıştır. Geçmiş ise o düğüm listenin başına yerleştirilmiştir .

Her adımdan sonra eklenen veya silinen bilgilerin kontrolü sağlanması için liste baştan sona ekrana yazdırılmıştır. Dosyadan okuma işlemi bittikten sonra da oluşturulan listenin silinip silinmeyeceği kullanıcıya sorulmuştur . Kullanıcı silinmesini istemiş ise liste tamamen sıfırlanmış ve kullanıcının yeni bir dosyadan okuma işlemi yapmak isteyip istemediği sorulmuştur . Eğer liste silinmek istenmez ise üstüne yeni dosyadan okunan değerler yazdırılabilmektedir .

### 3 - Karşılaşılan Sorunlar - - -

Tek sorun linkli liste kullanılmasından kaynaklı olarak düğümlere direkt bir erişim olmadığı için yapılan her bir ekleme silme ve değişiklik yapma işlemlerinde baş düğümden başlanarak istenen düğüm bulunana kadar her düğüme erişmek gerekiyor . Sürekli olarak aynı işlem yapılması gerekiyor . Algoritmayı küçük fonksiyonlara ayırarak ve uygun zamanlarda kontroller yapılarak her seferinden düğümlere erişmenin bir miktar önüne geçilebiliyor .

### 4 - Ekran Çıktıları - - -

#### ÖRNEK - 1

```
Add Info From File -> 1
Exit -> 0
->> 1
```

```
Enter the limit required to get to the first place -> 5
```

```
Enter the max number of nodes -> 4
```

```
Enter the file name (Enter -1 for Exit)-> address.txt
```

```
( A , 1 )
```

```
( B , 1 ) ----> ( A , 1 )
```

```
( B , 2 ) ----> ( A , 1 )
```

```
( B , 2 ) ----> ( A , 2 )
```

```
( C , 1 ) ----> ( B , 2 ) ----> ( A , 2 )
```

```
( C , 2 ) ----> ( B , 2 ) ----> ( A , 2 )
```

```
( C , 2 ) ----> ( B , 2 ) ----> ( A , 3 )
```

```
( C , 2 ) ----> ( B , 2 ) ----> ( A , 4 )
```

```
( C , 2 ) ----> ( B , 2 ) ----> ( A , 5 )
```

```
( A , 6 ) ----> ( C , 2 ) ----> ( B , 2 ) --->Eşik değeri aşıldığı an sınırı aşan
düğüm örnekte olduğu gibi (A=6). Bu
```

```
( A , 6 ) ----> ( C , 3 ) ----> ( B , 2 )
```

düğüm başa alınır .

**T = 5**

**L = 4**

Her yeni genel ilk sıraya yerleştirilir .

Aynı var ise sayaç arttırılır .

( D , 1 ) ----> ( A , 6 ) ----> ( C , 3 ) ----> ( B , 2 )	-----> Maksimum düğüm sayısına ulaşıldıktan sonra eklenecek yeni bir düğüm için sondaki düğüm de silinir .
<b>B silindi</b>	
( E , 1 ) ----> ( D , 1 ) ----> ( A , 6 ) ----> ( C , 3 )	
<b>C silindi</b>	
( F , 1 ) ----> ( E , 1 ) ----> ( D , 1 ) ----> ( A , 6 )	
<b>A silindi</b>	
( F , 1 ) ----> ( E , 1 ) ----> ( D , 2 ) ----> ( A , 6 )	
<b>A silindi</b>	
( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 ) ----> ( D , 2 )	
<b>D silindi</b>	
( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 ) ----> ( D , 3 )	
<b>D silindi</b>	
( B , 1 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	
<b>D silindi</b>	
( B , 1 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	

Bu algoritma için nasıl adresler verilirse verilirsin her türlü şekilde ekleme ve çıkarma işlemi yapılabilir . Silme işlemi için sadece soru sorulur eğer silinirse yeni bir liste oluşturulur . Aynı liste üzerinde işlem yapılmak istenirse devam ederek üstüne ekleme yapılır.

Enter the limit required to get to the first place -> 5	Aynı listeye yapılan ekleme işlemi
Enter the max number of nodes -> 4	
Enter the file name (Enter -1 for Exit)-> address.txt	
( A , 1 ) ----> ( B , 1 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	
( A , 1 ) ----> ( B , 2 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	
( A , 1 ) ----> ( B , 3 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	
( A , 2 ) ----> ( B , 3 ) ----> ( G , 1 ) ----> ( F , 1 ) ----> ( E , 1 )	

## ÖRNEK - 2

### ELLE KLAVYEDEN GİRME İŞLEMİ

-----  
MAIN MENU

```
| Add Info From File -> 1 |  
| Add Info From Keyboard -> 2 |  
| Change parameters (T L)-> 3 |  
| Exit -> 0 |  
-----
```

->> 2

Return for enter -> 0

Linked List Empty

Enter the address -> AA

( AA , 1 )

Enter the address -> BB

( BB , 1 ) ----> ( AA , 1 )

Enter the address -> CC

( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 1 )

Enter the address -> DD

( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 1 )

Enter the address -> AA

( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 2 )

Enter the address -> AA

( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 3 )

Enter the address -> AA

( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 4 )

Enter the address -> AA

( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 ) ----> ( AA , 5 )

Enter the address -> AA

( AA , 6 ) ----> ( DD , 1 ) ----> ( CC , 1 ) ----> ( BB , 1 )

Enter the address -> EE

( EE , 1 ) ----> ( AA , 6 ) ----> ( DD , 1 ) ----> ( CC , 1 )

Enter the address -> 0

( EE , 1 ) ----> ( AA , 6 ) ----> ( DD , 1 ) ----> ( CC , 1 )

Do you want to delete them all

Yes -> 1

No -> 2

->> 1

----- All Nodes Deleted Successfully -----

## ÖRNEK - 3

-----  
MAIN MENU

-----  
| Add Info From File -> 1 |  
| Add Info From Keyboard -> 2 |  
| Change parameters (T L)-> 3 |  
Exit -> 0

->> 1

Enter the file name (Enter -1 for Exit)-> address.txt

( A , 1 )

T = 1 , L = 1 için

( AA , 1 )

( A , 1 )

( A , 2 )

( B , 1 )

( BB , 1 )

( B , 1 )

( B , 1 )

( B , 2 )

( C , 1 )

( C , 2 )

( C , 3 )

( D , 1 )

( D , 2 )

( D , 3 )

Do you want to delete them all

Yes -> 1

No -> 2

->> 2