

Son Teslim tarihi: 17.03.2024 23:59

Ödev #1

OĞUZHAN TOPALOĞLU
Ç19052025 – Grup 1

*Bilgisayar Mühendisliği Bölümü,
Elektrik-Elektronik Fakültesi,
Yıldız Teknik Üniversitesi*



Istanbul, 2024

Video Linki: <https://www.youtube.com/watch?v=oK0jyWMjEkI>

İÇERİK

1. Problemin Tanımı
2. Problemin Çözümü
3. Karşılaşılan Sorunlar
4. Ekran Görüntüleri

1. Problemin Tanımı

Bu ödevde, websitelerinin tıklanma sayısı ve en son tıklanan websitelerin depolandığı bir cache buffer sisteminin tasarlanması istenilmiştir. Bu buffer sistemi tasarlanırken bizden doubly linkedlist veri yapısını kullanmamızı ve her node'un önceki ve sonraki node'ların adreslerinin tutmaları istenilmiştir.

Problemin zorluğu bize verilen T ve L değerlerinin implemente edilmesinde yatmaktadır. T değeri bize her node için bir tıklanma eşik değeri tanımlamaktadır. Eğer herhangi bir node'un tıklanma sayısı bu değeri geçerse bu node doubly linked list içerisinde yeni bir head olarak algılanmalı ve başa alınmalıdır. L değeri de bize buffer capacity limiti tanımlamaktadır. Eğer buffer içerisinde L sayısından çok website depolanırsa en sonda bulunan terimin silinmesi gerekmektedir.

2. Problemin Çözümü

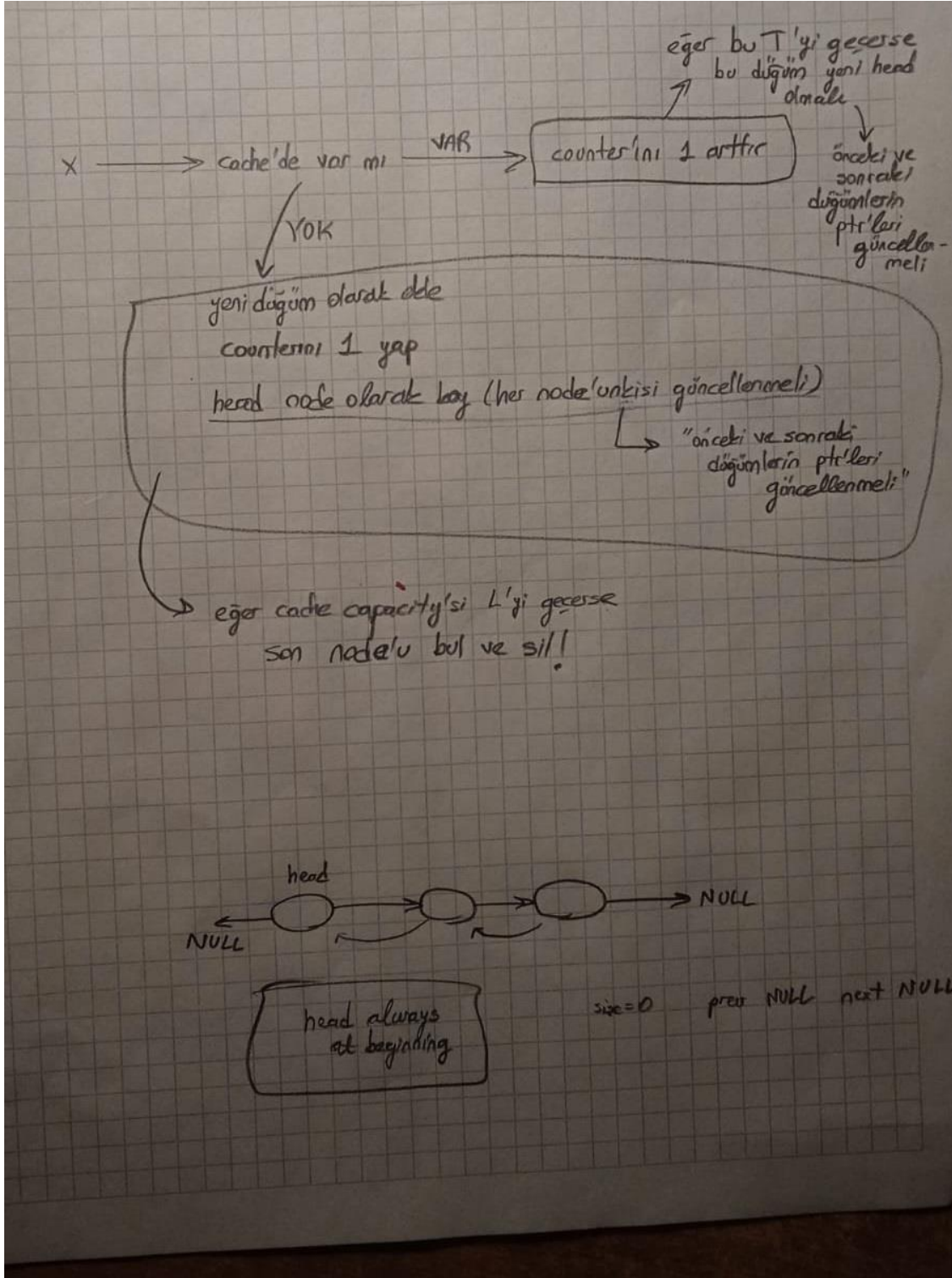
Yazılan kod hem kullanıcıdan hem de input.txt isimli dosyadan girdileri alarak void appendToList(Node**, int*, char*) fonksiyonunu çağırarak ve o anki doubly linkedlist düzenine göre eklemeler yapmaktadır. Kullanıcı isteği ile void printList(Node**) çağrılabilmekte ve listede bulunan her website, tıklanma sayıları ve adresleri ile birlikte terminalde yazdırılabilmektedir. Ayrıca yine kullanıcı isteği ile void clearList(Node**, int*) fonksiyonu çağrılabilmekte ve liste temizlenebilmektedir.

3. Karşılaşılan Sorunlar

Problemi çözerken en çok T ve L değerlerinin implementasyonlarında hata aldım.

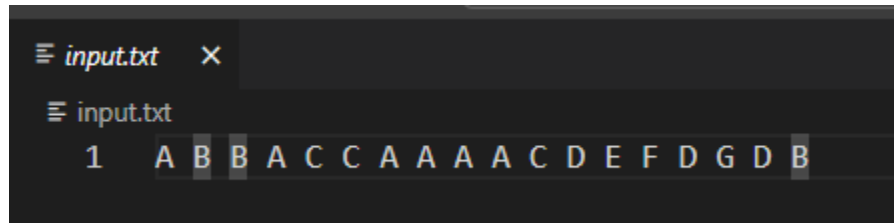
Karşılaştığım bug'lerden birisi şu idi: T=5, L=4 olsun, liste: A(5) => B(5) halinde iken eğer B gelirse, T değeri geçildiğinden B(6) => A(5) oluyordu ancak yine A gelirse T değeri geçildiğinden ve A çoktan head olduğundan dolayı NULL dolayısıyla hatalar oluşuyor ve segfault alıyordum.

Ayrıca problemi ilk okuduğumda çok anlamamıştım o yüzden elimle kağıt üzerinde şöyle bir flow-chart'ımsı bir çizim yaptım:



Bu çizim algoritmayı daha kolay anlamamı ve problemi 5-6 adet fonksiyona indirgeyebilmemi sağladı.

4. Ekran Görüntüleri



```
PS C:\Users\oguzh\Desktop\Kodlama\zSpecial\university-stuff\veri yapilari\hw1> .\run.bat
Could Not Find C:\Users\oguzh\Desktop\Kodlama\zSpecial\university-stuff\veri yapilari\hw1\C19052025.exe

C:\Users\oguzh\Desktop\Kodlama\zSpecial\university-stuff\veri yapilari\hw1>C19052025.exe

Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 1
Current list: A (1)
Current list: B (1) => A (1)
Current list: B (2) => A (1)
Current list: B (2) => A (2)
Current list: C (1) => B (2) => A (2)
Current list: C (2) => B (2) => A (2)
Current list: C (2) => B (2) => A (3)
Current list: C (2) => B (2) => A (4)
Current list: C (2) => B (2) => A (5)
Current list: A (6) => C (2) => B (2)
Current list: A (6) => C (3) => B (2)
Current list: D (1) => A (6) => C (3) => B (2)
Current list: E (1) => D (1) => A (6) => C (3)
Current list: F (1) => E (1) => D (1) => A (6)
Current list: F (1) => E (1) => D (2) => A (6)
Current list: G (1) => F (1) => E (1) => D (2)
Current list: G (1) => F (1) => E (1) => D (3)
Current list: B (1) => G (1) => F (1) => E (1)

Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: █
```

Current list: B (1) => G (1) => F (1) => E (1)

Choose option:

- 1- Enter from file
- 2- Enter manually
- 3- View list
- 4- Clear list
- 5- Quit

Selected: 2

Enter address as a character: G

Current list: B (1) => G (2) => F (1) => E (1)

Choose option:

- 1- Enter from file
- 2- Enter manually
- 3- View list
- 4- Clear list
- 5- Quit

Selected: 2

Enter address as a character: G

Current list: B (1) => G (3) => F (1) => E (1)

Choose option:

- 1- Enter from file
- 2- Enter manually
- 3- View list
- 4- Clear list
- 5- Quit

Selected: 2

Enter address as a character: G

Current list: B (1) => G (4) => F (1) => E (1)

Choose option:

- 1- Enter from file
- 2- Enter manually
- 3- View list
- 4- Clear list
- 5- Quit

Selected:

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 2
Enter address as a character: G
Current list: B (1) => G (4) => F (1) => E (1)
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 2
Enter address as a character: G
Current list: B (1) => G (5) => F (1) => E (1)
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 2
Enter address as a character: G
Current list: G (6) => B (1) => F (1) => E (1)
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: █
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 3
Current list: G (6) => B (1) => F (1) => E (1)
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 4
List was succesfully deleted.
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: 3
Current list: <empty>
```

```
Choose option:
1- Enter from file
2- Enter manually
3- View list
4- Clear list
5- Quit
Selected: █
```