

# 2019-2020 Bahar Yarıyılı BLM2512 Veri Yapıları ve Algoritmalar 1. Ödevi

Mesut Şafak Bilici  
17011086

Öncelikle bu ödevde, *Doubly Linked List* kullanılacağı için yapıyı hazırlamam gerekti. Bunun için toplamda 6 fonksiyon kullandım:

- void push(NODE\*\*,char\*) : Yeni eleman geldiğinde en başa atan.
- void del(NODE\*\*,NODE\*) : Linked List elemanı silen
- void print(NODE\*\*) : Linked List'i bastıran.
- void push2(NODE\*\*,char\*,int) : Aynı elemandan geldiğinde fakat network eşliğini geçtiğinde başa atan.
- void clear(NODE\*\*) : Tüm Linked List elemanlarını silen.
- void algorithm(char\*,NODE\*,int,int,int,int) : Networkler scanf ile alınırsa çalışacak algoritma.

Bu fonksiyonların iç kodları kod içinde yazıldı ve içinde tekrardan detaylıca açıklandı.

Daha sonra algoritmanın yapısı oluşturuldu. Algoritma temelde üç durumdan ve bu durumlar birçok alt durumdan oluşmaktadır:

1. Linked List ilk atanmaya başlandığında **head == NULL** olma durumu.
2. Linked List'in eşik değerinin altında olduğu durum.
  - (a) Gelen yeni string'in Linked List'te olmama durumu veya sondaki elemanla eşleşmesi.
    - current next = NULL olduğu an ya string yoktur ya da string en sonda eşleşmiştir.

- Gelen network eşleştirse, eşleştikten sonra T değerinin eşiği geçmesi ya da geçmemesi.
- (b) Gelen yeni string'in Linked List'te olma durumu.
- 3. Linked List'in eşik değerinin üstünde olduğu durum.
  - (a) Gelen yeni string'in Linked List'te olmama durumu veya sondaki elemanla eşleşmesi.
    - current next = NULL olduğu an ya string yoktur ya da string en sonda eşleşmiştir.
      - Gelen network eşleştirse, eşleştikten sonra T değerinin eşiği geçmesi ya da geçmemesi.
  - (b) Gelen yeni string'in Linked List'te olma durumu.
    - Gelen network eşleştirse, eşleştikten sonra T değerinin eşiği geçmesi ya da geçmemesi.

#### **Çalıştırmaya Dair Bazı Notlar:**

- Eğer ki terminal üzerinden argüman vererek çalıştıracaksanız dosyanızın içinden T ve L değerlerini siliniz.
  - argv[0]: ./17011086 — argv[1]: yourfilename.txt — argv[2]: T — argv[3]: L.
- Her şeyi file'dan okuyacaksanız argüman seçmeyip ilk sorulan soruya gerekli inputu giriniz. Dosyanın ilk iki satırında bu kez T ve L değeri olmak zorunda.
  - **Örnek (input1.txt):**
    1. 2
    2. 3
    3. AB...
- Eğer ki her şeyi stdin ile almak istiyorsanız, file kullanmanıza gerek yok. Alımı durdurmak istiyorsanız "Network ismi giriniz" yazısından sonra EOF sinyali göndererek alımı durdurabilirsiniz (CTRL + D).

**Aşamalı ve Detaylı Açıklanması:** Program başladığı zaman sizden 3 seçenekli bir input bekliyor.

1. Argüman ile çalıştırma: 1
2. File ile çalıştırma: 0
3. Scanf ile çalıştırma: -1

Argüman olarak çalıştırınca da arkada file kullanıyor yukarıdaki gibi belirtildiği gibi. Argümanları aldıktan sonra dosyamızın içinde ilk iki satırında T ve L değerleri olmamalı. Dosyadan stringleri alırken fgets fonksiyonunu kullandım ve bu line'ların sonundaki newline karakterini string tokenizing yöntemleri ile ayırdım. Daha sonra yukarıda belirtilen koşullu ifadeleri kontrol ederek algoritmamı düzenledim. File ile çalıştırınca ilk iki satırda T ve L değerleri olmalı input dosyamızda. Scanf ile çalıştırınca file gerekmiyor. Size T ve L değerlerini soracak program. Daha sonra siz EOF sinyali gönderene kadar input istemeye devam edecektir.

Bu durumlar göze alınarak birçok iç içe koşullu yapılardan sonra program ödevin doğru çıktılarını aşağıdaki gibi vermektedir:

<i>Girdi</i>	AB	BA	CY	AB	CY	XYZ	BA	XYZ	BA
--------------	----	----	----	----	----	-----	----	-----	----

Çıktı:

```
safak@progcu:~/Desktop/BLM2512 Homework1$ ./doubly input1.txt 2 3
Do you want to execute this program with using main arguments? If yes PRESS 1, else PRESS 0:1
* AB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BA,1 - AB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* CY,1 - BA,1 - AB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* CY,1 - BA,1 - AB,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* CY,2 - BA,1 - AB,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* XYZ,1 - CY,2 - BA,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* XYZ,1 - CY,2 - BA,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* XYZ,2 - CY,2 - BA,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BA,3 - XYZ,2 - CY,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
Q.E.D
safak@progcu:~/Desktop/BLM2512 Homework1$
```

<i>Girdi</i>	A	B	A	AA	BBB	B	A	AB	A	B	A	BB
--------------	---	---	---	----	-----	---	---	----	---	---	---	----

Çıktı:

```
safak@progcu:~/Desktop/BLM2512 Homework1$ ./doubly input2.txt 3 4
Do you want to execute this program with using main arguments? If yes PRESS 1, else PRESS 0:1
* A,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* B,1 - A,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* B,1 - A,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* AA,1 - B,1 - A,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BBB,1 - AA,1 - B,1 - A,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BBB,1 - AA,1 - B,2 - A,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BBB,1 - AA,1 - B,2 - A,3
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* AB,1 - BBB,1 - AA,1 - B,2
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* A,1 - AB,1 - BBB,1 - AA,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* B,1 - A,1 - AB,1 - BBB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* B,1 - A,2 - AB,1 - BBB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0
* BB,1 - B,1 - A,2 - AB,1
Do you want to clear that all the network buffer? If it yes PRESS 1, else PRESS 0: 0

Q.E.D
safak@progcu:~/Desktop/BLM2512 Homework1$
```

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 100
struct node{
    char net[MAX];
    struct node *next,*prev;
    int T;
};
typedef struct node NODE;
void push(NODE**,char*);
void del(NODE**,NODE*);
void print(NODE**);
void push2(NODE**,char*,int);
void clear(NODE**);
void algorithm(char*, NODE*, int, int , int , int);
int main(int argc, char **argv) {
    int argument;
    int a=0;
    NODE* head = NULL; // creating the null

    printf("Do you want to execute this program with using main arguments or file or stdin? If arguments PRESS 1, if file PRESS 0, if stdin press -1: ");
    scanf("%d",&argument);
    int T; // the threshold of single network's number.
    int L; // the capacity of the doubly linked list.
    char* file; // the file that we'll open.
    static int controllerL = 0; // controller of length of our doubly linked list.

    FILE* fp;
    char* buffer = (char*)calloc(MAX,sizeof(char)); // stores our network string
    .
    int _clear; // controller of whether the user wants to delete all linked list.

    if(argument == 1) {
        if(argc != 4) { // argument number controller.
            fprintf(stderr,"ERROR: Argument Error!\n");
            printf("Usage: || argv[0]: ./executable_file_name || argv[1]: input_file_name .txt || argv[2]: T || argv[3]: L ||\n");
            exit(EXIT_FAILURE);
        }
        fp = fopen(argv[1],"r"); //opening file.
        if(fp == NULL) {
            exit(EXIT_FAILURE);
        }

        T=atoi(argv[2]);
        L=atoi(argv[3]);
        printf("\n");
    }
    else if(argument==0){
        file = (char*)malloc(MAX*sizeof(char)); // file string buffer
        printf("Enter the file name: ");
        scanf("%s",file);
        fp = fopen(file,"r");
        fgets(buffer,MAX-1,fp);
        T=atoi(strtok(buffer,"\n"));
        fgets(buffer,MAX-1,fp);
        L=atoi(strtok(buffer,"\n"));
        printf("T: %d\n",T);
        printf("L: %d\n",L);
    }
    else{
        printf("T,L: ");
        scanf("%d %d",&T,&L);
        printf("\n");
        algorithm(buffer,head,controllerL,T,L,_clear);
        return 0;
    }

    // scanning the file, line by line.
    while(fgets(buffer,MAX-1,fp)) {
        NODE* current = head;

```

```

        buffer = strtok(buffer, "\n");
        // when you read the file line by line, line buffer has "\n" before
the "\0". To get rid of it we use strtok.
        // when we print the stringname, networknumber, the network number is
printed in new line
        // due to the string has "\n". strtok helps us in this situation.
        if(head==NULL){ // for first node of doubly linked list.
            push(&head,buffer);
            controllerL++;
        }
        else if(controllerL < L) { // if linked list length lower than the c
apacity...
            while(current->next != NULL && strcmp(current->net,buffer) !=
= 0){
                current = current->next;
            }
            if(current->next == NULL) {
                // if the last node's net is equal to string buffer
and current->next is NULL.
                if(strcmp(current->net,buffer) == 0) {
                    if(current->T == T) {
                        current->T++;
                        push2(&head,buffer,current->T);
                        del(&head,current);
                    }
                    else{
                        current->T++;
                    }
                }
                else{
                    push(&head,buffer);
                    controllerL++;
                }
            }
            // if linked list length equals or large than the capacity..
.
            else{
                if(current->T == T) {
                    current->T++;
                    push2(&head,buffer,current->T);
                    del(&head,current);
                }
                else{
                    current->T++;
                }
            }
        }
        else{
            while(current->next != NULL && strcmp(current->net,buffer) !=
= 0) {
                current=current->next;
            }
            if(current->next == NULL) {
                if(strcmp(current->net,buffer) == 0) {
                    if(current->T == T) {
                        current->T++;
                        push2(&head,buffer,current->
T);
                        del(&head,current);
                    }
                    else{
                        current->T++;
                    }
                }
                else{
                    push(&head,buffer);
                    del(&head,current);
                }
            }
            else{
                printf("does it even work");

```

```

        if(current->T == T) {
            current->T++;
            push2(&head,buffer,current->T);
            del(&head,current);
        }
        else{
            current->T++;
        }
    }
    print(&head);
    printf("\n");
    printf("Do you want to clear that all the network buffer? If it yes
PRESS 1, else PRESS 0: ");
    scanf("%d",&_clear);
    printf("\n");
    if(_clear == 1) {
        clear(&head);
        controllerL = 0;
    }
    _clear=0;
}
return 0;
}

/*
* The difference between push and push2 function is to
* determine that whether current->T > T or the buffer is new.
* if current->T > T, function push2 push the buffer to the front.
* if the buffer is new, function push the buffer to the front and make new node's T
= 1.
*/
void push(NODE** head,char* buffer) {
    NODE* newnode = (NODE*)malloc(sizeof(NODE));
    if(*head==NULL) { // for initialize head
        newnode->next = NULL;
        newnode->prev = NULL;
        strcpy(newnode->net,buffer);
        newnode->T=1;
        (*head) = newnode;
    }
    else{
        strcpy(newnode->net,buffer);
        newnode->prev = NULL;
        newnode->next = (*head);
        (*head)->prev = newnode;
        (*head) = newnode;
        newnode->T=1;
    }
}

void del(NODE** head,NODE* del){
    if(del->next == NULL){ // whether the del node head or the last node of link
ed list.
        if((*head)->next == NULL) {
            head=NULL;
            free(head);
        }
        else{
            del->prev->next=NULL;
            free(del);
        }
    }
    else{ // if the del node inside the linked list.
        del->prev->next = del->next;
        del->next->prev = del->prev;
    }
}

void print(NODE** head) {

```



```

    NODE* current = *head;
    printf("* ");
    while(current->next != NULL) {
        printf("%s,%d - ", current->net, current->T);
        current=current->next;
    }
    printf("%s,%d\n", current->net, current->T);
}

void push2(NODE** head, char* buffer, int T) {
    NODE* newnode = (NODE*)malloc(sizeof(NODE));
    if(*head==NULL) {
        newnode->next = NULL;
        newnode->prev = NULL;
        strcpy(newnode->net, buffer);
        newnode->T=T;
        (*head) = newnode;
    }
    else{
        strcpy(newnode->net, buffer);
        newnode->prev = NULL;
        newnode->next = (*head);
        (*head)->prev = newnode;
        (*head) = newnode;
        newnode->T=T;
    }
}

void clear(NODE** head) {
    NODE* current = *head;
    NODE* next;
    while(current != NULL) {
        next = current->next;
        free(current);
        current=next;
    }

    *head = NULL;
}

void algorithm(char* buffer, NODE* head, int controllerL, int T, int L, int _clear) {
    printf("Enter network name: ");
    while(scanf("%s", buffer) != EOF) { // scanning the file, line by line.
        printf("\n");
        NODE* current = head;
        if(head==NULL){ // for first node of doubly linked list.
            push(&head, buffer);
            controllerL++;
        }
        else if(controllerL < L) { // if linked list length lower than the capacity...
            while(current->next != NULL && strcmp(current->net, buffer) != 0) {
                current = current->next;
            }
            if(current->next == NULL) {
                // if the last node's net is equal to string buffer
                and current->next is NULL.
                if(strcmp(current->net, buffer) == 0) {
                    if(current->T == T) {
                        current->T++;
                        push2(&head, buffer, current->T);
                        del(&head, current);
                    }
                    else{
                        current->T++;
                    }
                }
            }
            else{

```

```

    push(&head,buffer);
    controllerL++;
}

    }
    else{ // if linked list length equals or large than the capa
city...
        if(current->T == T) {
            current->T++;
            push2(&head,buffer,current->T);
            del(&head,current);
        }
        else{
            current->T++;
        }
    }
}
else{
    while(current->next != NULL && strcmp(current->net,buffer) !=
= 0) {
        current=current->next;
    }
    if(current->next == NULL) {
        if(strcmp(current->net,buffer) == 0) {
            if(current->T == T) {
                current->T++;
                push2(&head,buffer,current->
T);
                del(&head,current);
            }
            else{
                current->T++;
            }
        }
        else{
            push(&head,buffer);
            del(&head,current);
        }
    }
    else{
        printf("does it even work");
        if(current->T == T) {
            current->T++;
            push2(&head,buffer,current->T);
            del(&head,current);
        }
        else{
            current->T++;
        }
    }
}
print(&head);
printf("\n");
printf("Do you want to clear that all the network buffer? If it yes
PRESS 1, else PRESS 0: ");
scanf("%d",&_clear);
printf("\n");
if(_clear == 1) {
    clear(&head);
    controllerL = 0;
}
_clear=0;
printf("Enter network name: ");
}
}

```