

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



2018-2019 BAHAR YARIYILI
BLM2512 VERİ YAPILARI VE ALGORİTMALAR DERSİ
GRUP-2
DÖNEM PROJESİ RAPORU

Hazırlayan: Mehmet Hayri Çakır – 16011023

Dersin Yürütücüsü: Dr.Öğr.Üyesi Göksel Biricik

İstanbul, Mayıs 2019

İçindekiler

1. Yöntem.....	2
1.1. Problem Tanımı.....	2
1.2. Akış Diyagramları	2
1.2.1. enqueue() ve dequeue().....	2
1.2.2. getInputFile() ve testGraphValidity().....	3
1.2.3. populateAdjacencyMatrix() ve findWordsInFile()	4
1.2.4. findTransformation()	5
2. Uygulama	6
2.1. Dönüşüm Olan Örnekler	6
2.1.1. baron-bison kelimeleri arası dönüşüm	6
2.1.2. withy-witty kelimeleri arası dönüşüm	7
2.2. Dönüşüm Olmayan Örnek.....	7
2.2.1. worry-worse kelimeleri arasında dönüşüm yoktur.....	7
3. Dipnot.....	8

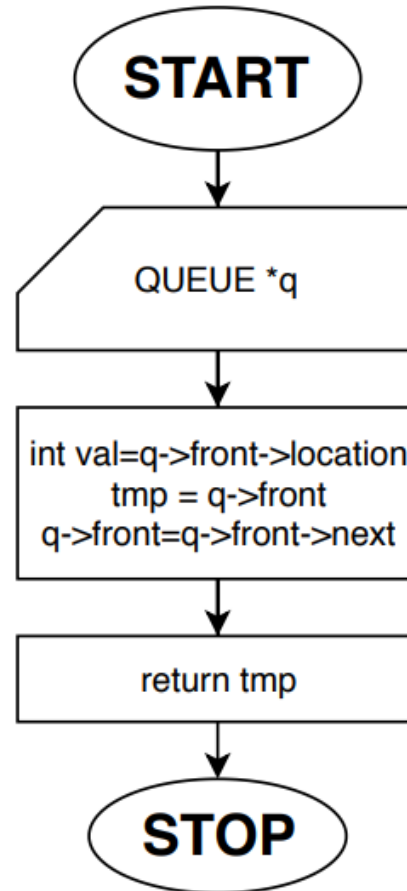
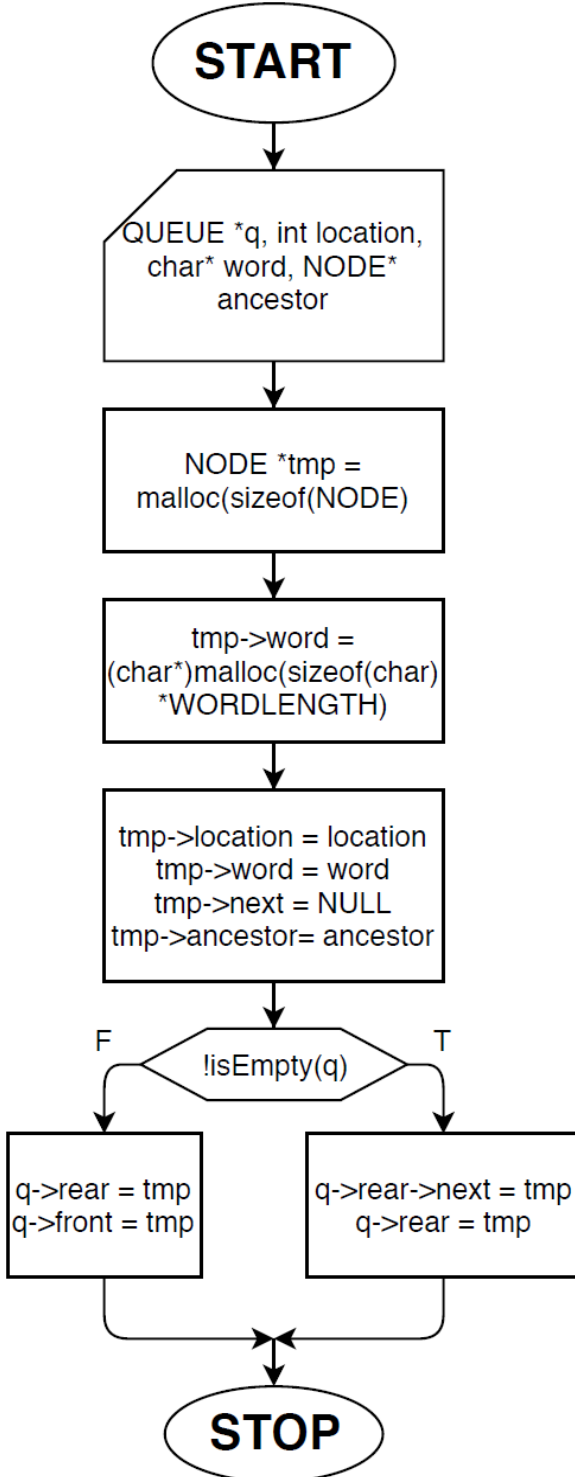
1. Yöntem

1.1. Problem Tanımı

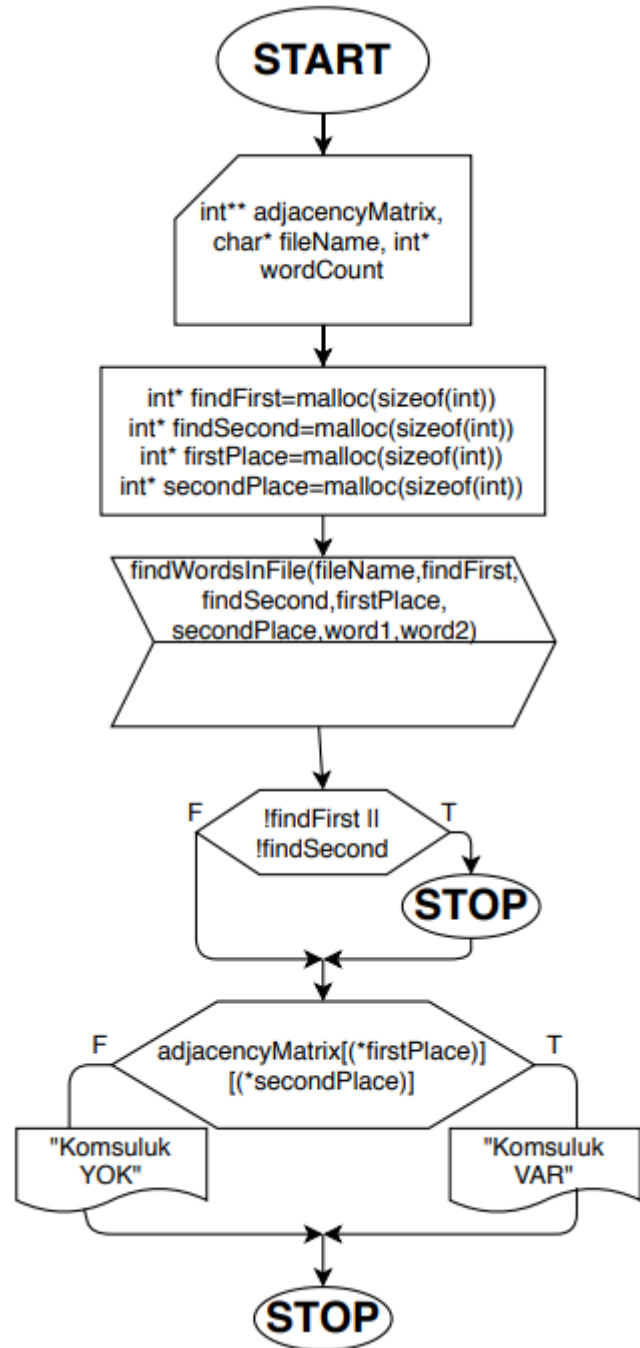
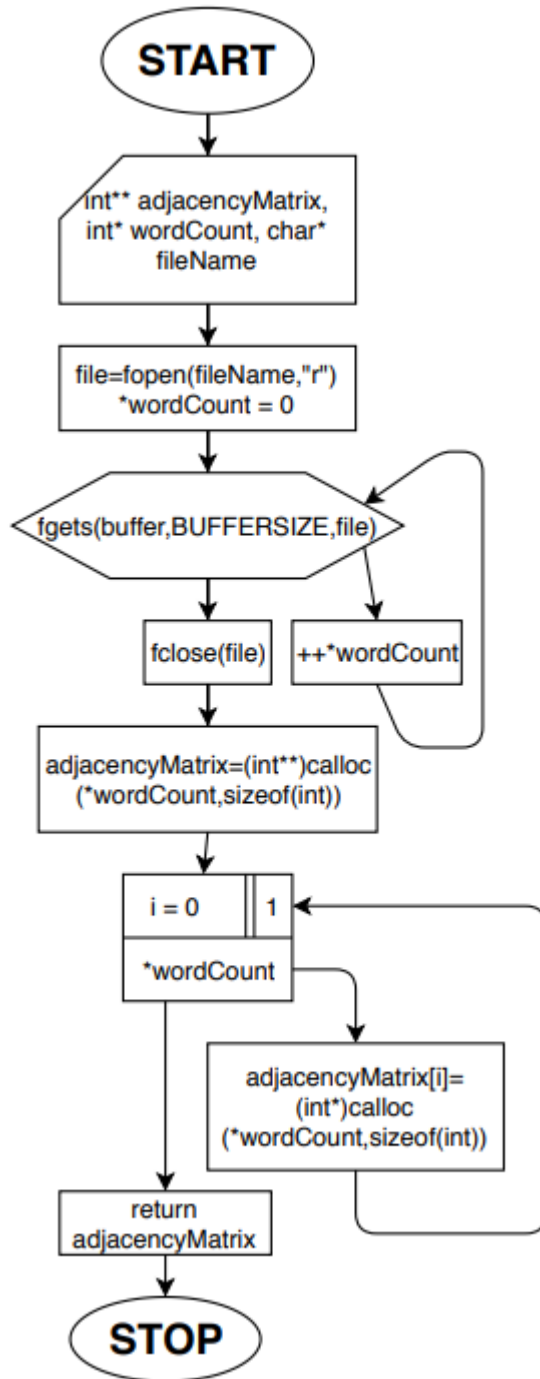
Bu projede bizden istenen şey temelinde word ladder problemi çözümü olmakla birlikte, tüm kuyruğun yanında tek harf dönüşümü içeren yolu da yazdırmamızdır. Bu problemi çözerken BFS algoritmasına ek olarak her bir node'un atasını da tutarak yolu yazdırdım. Aynı zamanda kuyruktan çıkan kelimeleri de yazdırdım çünkü ödev metninde istenen buydu ilk başta, kodumu buna göre yazmışken deadline'a 2 gün kala dönüşüm yolunu yazdırmamız istendi. Ne istendiği net olmadığı için 2 türlü de ekrana yazdırıyorum.

1.2. Akış Diyagramları

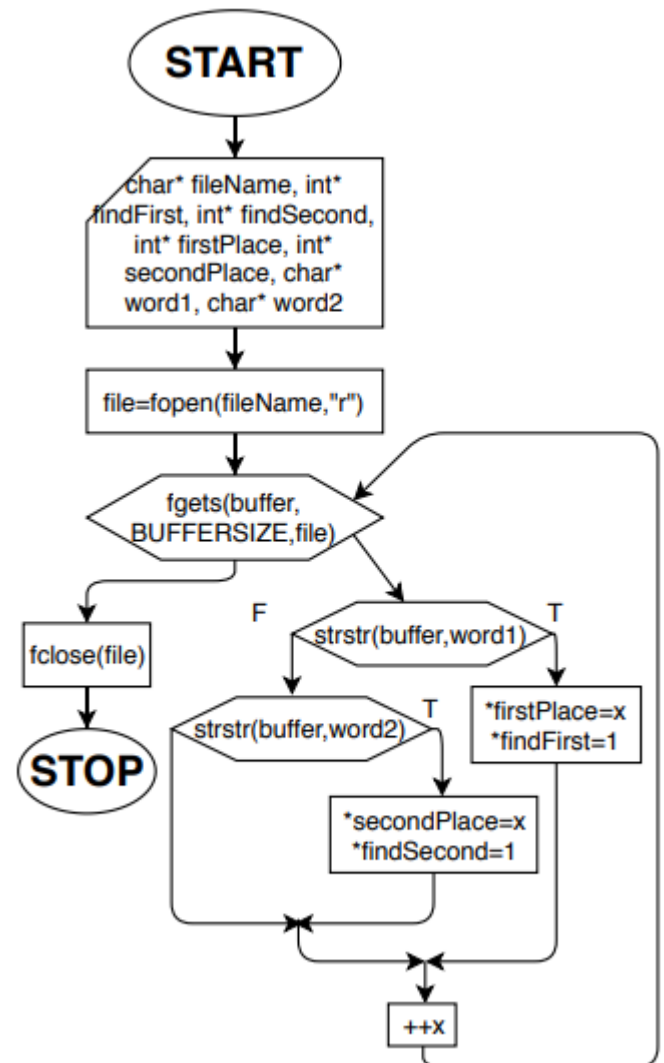
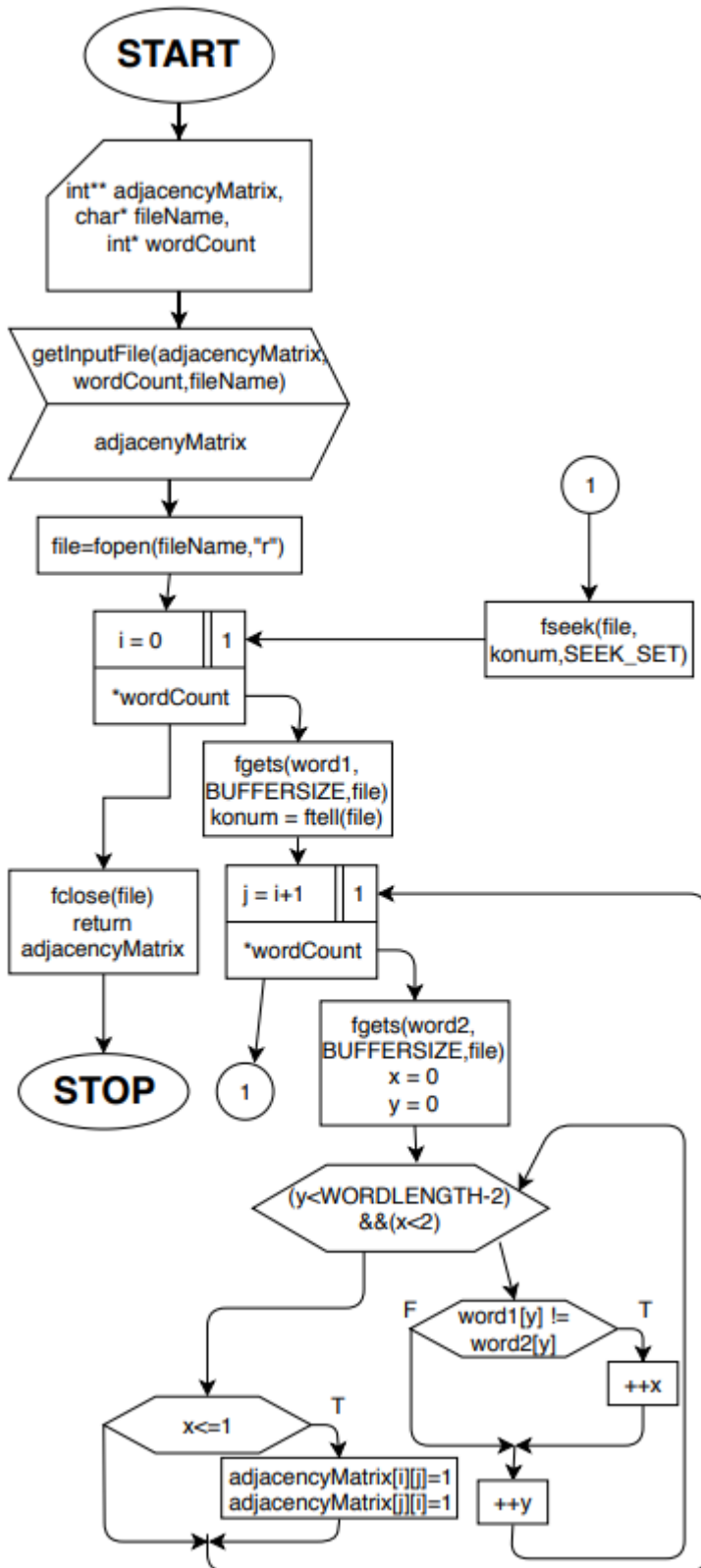
1.2.1. enqueue() ve dequeue()



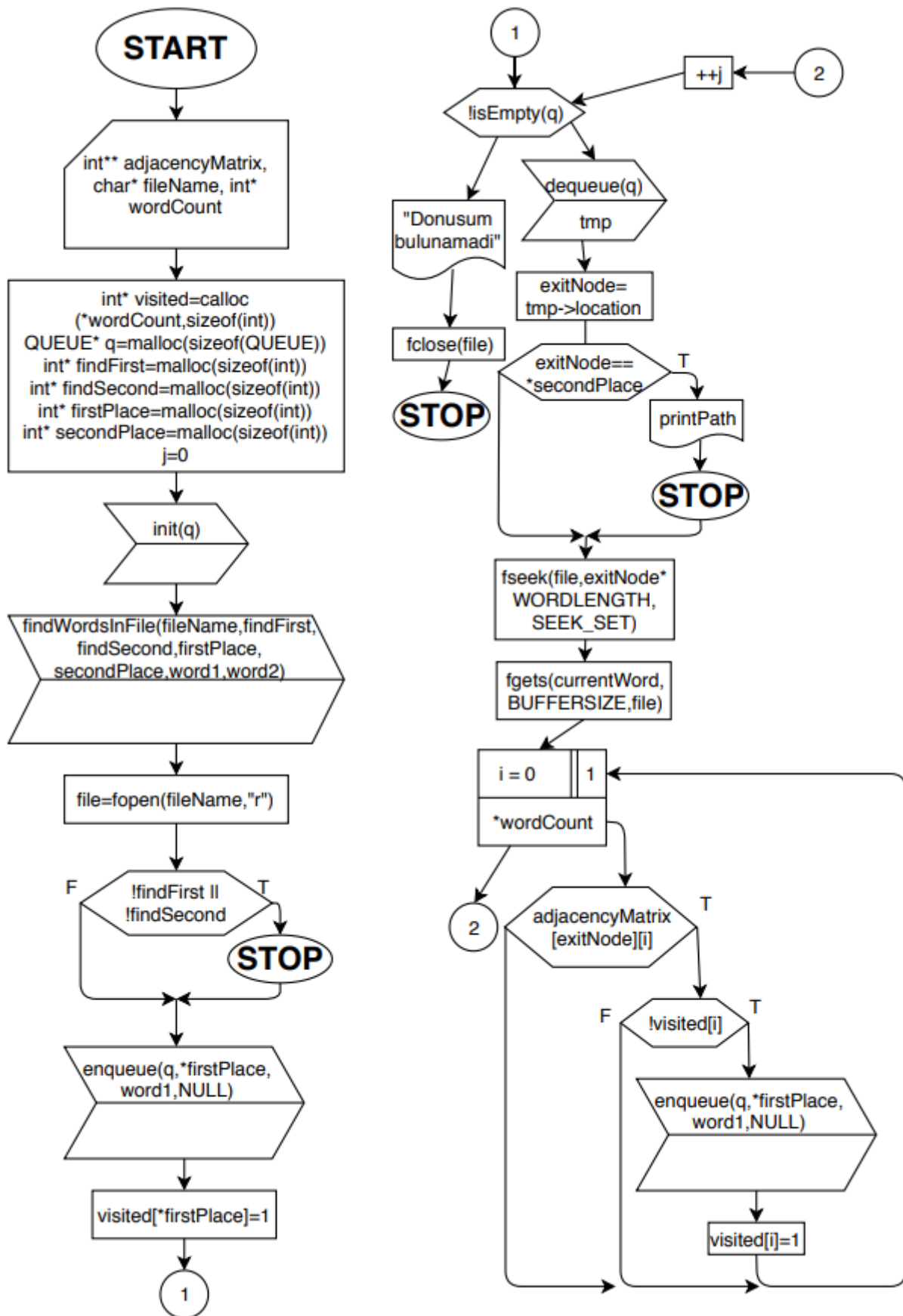
1.2.2. getInputFile() ve testGraphValidity()



1.2.3. populateAdjacencyMatrix() ve findWordsInFile()



1.2.4. findTransformation()



2. Uygulama

2.1. Dönüşüm Olan Örnekler

2.1.1. baron-bison kelimeleri arası dönüşüm

Kuyruğa baron atılır.

Kuyruk: baron

Kuyruktan baron çekilir, komşuları atılır.

Kuyruk: bacon-baton-boron

Kuyruktan sırasıyla bacon-baton-boron çekilir.

Kuyruk: boron

Kuyruktan boron çekilir, komşuları atılır.

Kuyruk: boson-moron

Kuyruktan boson çekilir, komşuları atılır.

Kuyruk: moron-bison

Kuyruktan moron-bison çekilir. bison'a ulaştığımız için işlem tamamlandı.

```
1. kelimeyi giriniz: baron
2. kelimeyi giriniz: bison
Tum kuyruk yazdiriliyor..
    1-baron
    2-bacon
    3-baton
    4-boron
    5-boson
    6-moron

Yol yazdiriliyor..
    0-baron
    1-boron
    2-boson
    3-bison

>>>Islem 3 adimda tamamlandi.
bison bulundu.
```

2.1.2. withy-witty kelimeleri arası dönüşüm

Kuyruğa withy atılır.

Kuyruk: withy

Kuyruktan withy çekilir ve komşuları atılır.

Kuyruk: pithy-wishy-withe-witty

Kuyruktan pithy-wishy-withe çekilir girecek komşuları yok atılmaz.

Kuyruk: witty

Kuyruktan witty çekilir ve işlem tamamlanır.

```
1. kelimeyi giriniz: withy
2. kelimeyi girinz: witty
Tum kuyruk yazdiriliyor..
    1-withy
    2-pithy
    3-wishy
    4-withe

Yol yazdiriliyor..
    0-withy
    1-witty

>>>Islem 1 adimda tamamlandi.
witty bulundu.
```

2.2. Dönüşüm Olmayan Örnek

2.2.1. worry-worse kelimeleri arasında dönüşüm yoktur.

Kuyruğa worry atılır.

Kuyruk: worry

Kuyruktan worry çekilir, komşuları atılır.

Kuyruk: sorry-wordy-wormy

Kuyruktan sorry çekilir komşusu yok, wordy çekilir komşuları atılır.

Kuyruk: wormy-woody

Kuyruktan wormy çekilir komşusu yok, woody çekilir komşuları atılır.

Kuyruk: goody-moody

Kuyruktan goody ve moody çekilir komşuları yok, kuyruk boşaldı dönüşüm yoktur.

```
1. kelimeyi giriniz: worry
2. kelimeyi girinz: worse
Tum kuyruk yazdiriliyor..
    1-worry
    2-sorry
    3-wordy
    4-wormy
    5-woody
    6-goody
    7-moody
Donusum bulunamadi.
```


3. Dipnot

DevC++'ta do while yapısı çalışmıyor ve program her 2 kelimeyi kontrol ettikten sonra kapanıyor, sebebini bulamadım. Bu yüzden her 2 kelimeyi kontrol ettikten sonra kapatıp açmanız gerekebilir.