Kelime Merdiveni Oyunu

Öğrenci Adı: Burak Atalay

Öğrenci Numarası: 22011641

Dersin Eğitmeni: Göksel Biricik

Video Linki: https://youtu.be/fhc221_epKU

1- Problem Tanımı:

Bu ödevde kullanıcıdan alınan kaynak ve hedef kelimeleri arasında bir kelime merdiveni bulunması isteniyor. Bu merdiven, dosyadan okunan sözlük içerisinden şartları sağlayan kelimeler seçilerek oluşturulur. Şartlar ise merdivendeki kelimelerin aynı uzunlukta olması ve merdivendeki komşu kelimeler arasında yalnızca bir harfin farklı olmasıdır.

2- Problemin Çözümü:

Problemde kullanılacak stack ve queue yapıları linkli liste ile implemente edildi. Kullanıcıdan alınan kaynak ve hedef kelimeleri aynı uzunlukta değilse program sonlanır. Aynı ise kullanıcının sağladığı dosyadan kelimeler alınır ve kaynak ile aynı uzunluktakiler stack yapısındaki sözlüğe kaydedilir. Sonrasında problemde verilen algoritmanın implemente edildiği findWordLadder fonksiyonu çağrılır.

Aslında problem tanımında verilen algoritma bir BFS implementasyonu çünkü kaynak kelimeden başlanıp bir uzaklıktaki yani bir harf farkı olan tüm kelimeler bulunup queue'ya ekleniyor. Yani aslında bu graph gibi düşünülürse kaynak kelimenin tüm komşuları bulunup queue'ya ekleniyor ve bir dahaki iterasyonda queue'nun başındaki stack alınıp onun başındaki kelimenin komşuları işleme alınıyor. Buna devam ederken işlenmiş kelimeleri bir daha işleme almamak için o kelimelerin işaretlenmesi gerekiyor. Eğer problemin çözüm yolunu kaynak kelimenin root olduğu bir ağaç gibi düşünürsek, işlem yapılan kelimenin işaretlenmediği durumda bir kelime ya kendi seviyesinden bir kelime ile ya da bir üst seviyedeki parent'ı haricindeki bir kelime ile komşuluk kurmuş olur ve bu da merdiveni gereksiz şekilde uzatır. Bir başka durum ise bir alt seviyedeki önceden işlenip işaretlenmiş bir kelimenin üst seviyedeki başka kelimeler tarafından eklenememesidir. Bu durumda zaten işaretlenmiş kelime queue'da olduğundan onun komşuları gezilecektir. Bu yüzden üst seviyedeki başka bir kelime tarafından eklenmesine gerek yoktur.

Fonksiyonun içeriğine gelinirse, önce tüm sözlük gezilip kaynak kelime bulunur, işaretlenir ve stack oluşturulup queue'ya eklenir. İşaretleme için STACK struct'ında isVisited isimli 1 bitlik unsigned int değişkeni var. Bu değişken sadece sözlük için kullanılır, queue'ya eklenen stack'larda bu değişken var olsada kullanılmaz. Do while döngüsüne girdikten sonra dequeue işlemi yapılıp alınan stack'in başındaki kelime hedef kelime ile aynı değilse bu kelimenin komşuları yani bir harf farklı olan kelimeler sözlükte aranır, eğer önceden queue'ya eklenmedilerse dequeue edilen stack kopyalanır ve bu kelime kopyalanmış stack'e push edilir. Yani aslında BFS'de olduğu gibi o anki node'un işlenmemiş komşusu bulunup o node'un enqueue edilme işlemi yapılıyor. Tek fark, sadece o kelimenin enqueue edilmesi yerine kaynak kelimeden o kelimeye kadarki merdivenin bulunduğu stack'in enqueue edilmesi. Stack kopyalama kısmında merdivenin yapısının korunması için kopyalanan stack yardımcı bir stack yardımıyla ters çevrilip ondan sonra ters çevrilmiş stack yeni stack'e head'den başlayarak push ediliyor. Eğer döngünün başında dequeue edilen stack'teki kelime hedef kelime ile aynıysa fonksiyondaki control değişkeni 1 yapılır. Döngü queue boş olmadığı ve control değikeni 0 olduğu sürece devam eder. control 1 olmuşsa merdiven tamamlanmıştır. Queue boşalmışsa tüm kelimeler gezilmiş ancak merdiven tamamlanamamış demektir.

3- Karşılaşılan Sorunlar:

Ödevde verilen sözlük boyutu çok fazla olduğundan iki boyutlu karakter array'i için statik bir yer açma girişiminde program patlıyordu. Bu sebeple sözlüğü tutmak için zaten linki liste ile implemente ettiğim stack yapısını kullanmaya karar verdim. Aslında sözlüğü stack olarak değil de linkli liste olarak kullanıyorum.

Ayrıca işlem yaptığım kelimeleri sözlükte işaretlemek için struct'ta bir değişken tutmak, işlem yapılmış mı kontrolünü O(1) karmaşıklığa indirdi. İlk başta işlem yapılan kelimeleri ayrı bir linkli listeye almayı düşünyordum ancak bu O(n)'lik bir karmaşıklık getiriyordu. Yani do while içindeki bir iterasyon için toplam karmaşıklık en kötü durumda O(n^2) olacaktı. Ancak şu anda O(n).

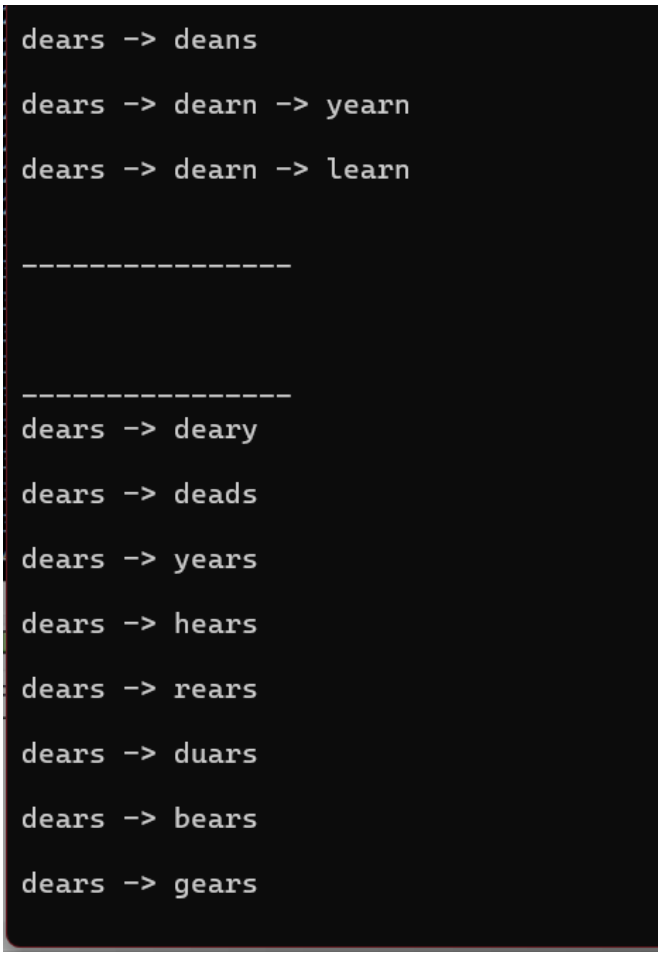
4- Ekran Çıktıları:

Senaryo 1-

```
source word: dears
destination word: fears
filename: dictionary.txt

----------------
dears -> dearn
dears -> fears
dears -> deary
dears -> deads
dears -> years
dears -> hears
dears -> rears
dears -> duars
dears -> bears
dears -> gears
dears -> deare
```

```
dears -> lears
dears -> sears
dears -> deers
dears -> tears
dears -> wears
dears -> pears
dears -> nears
dears -> deals
dears -> deans

----------------


----------------
dears -> fears
dears -> deary
dears -> deads
```

```
dears -> years
dears -> hears
dears -> rears
dears -> duars
dears -> bears
dears -> gears
dears -> deare
dears -> lears
dears -> sears
dears -> deers
dears -> tears
dears -> wears
dears -> pears
dears -> nears
dears -> deals
```


dears -> deans
dears -> dearn -> yearn
dears -> dearn -> learn
dears -> deary
dears -> deads
dears -> years
dears -> hears
dears -> rears
dears -> duars
dears -> bears
dears -> gears

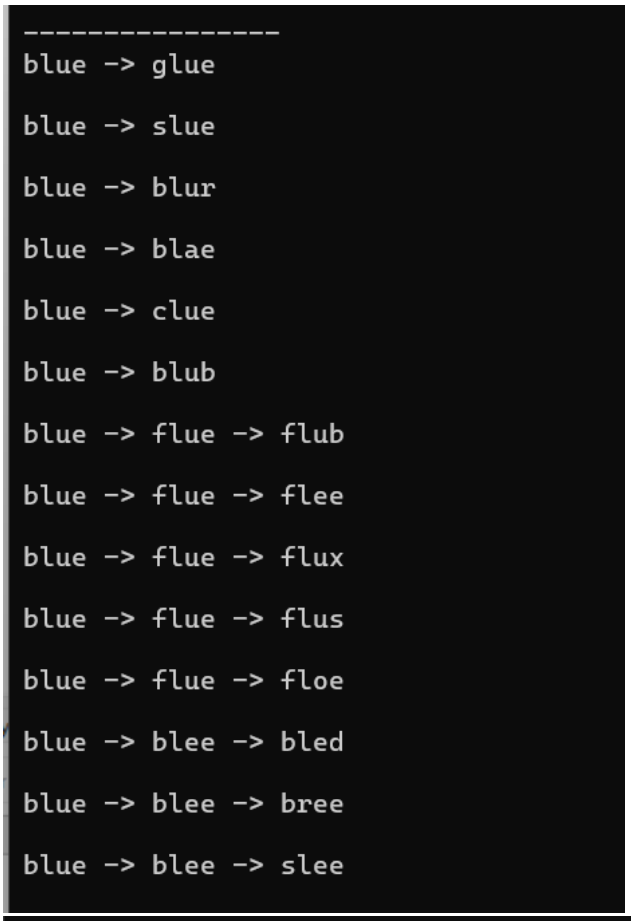

dears -> deare
dears -> lears
dears -> sears
dears -> deers
dears -> tears
dears -> wears
dears -> pears
dears -> nears
dears -> deals
dears -> deans
dears -> dearn -> yearn
dears -> dearn -> learn
dears -> fears
Process exited after 4.471 seconds with return value 0
Press any key to continue . . .

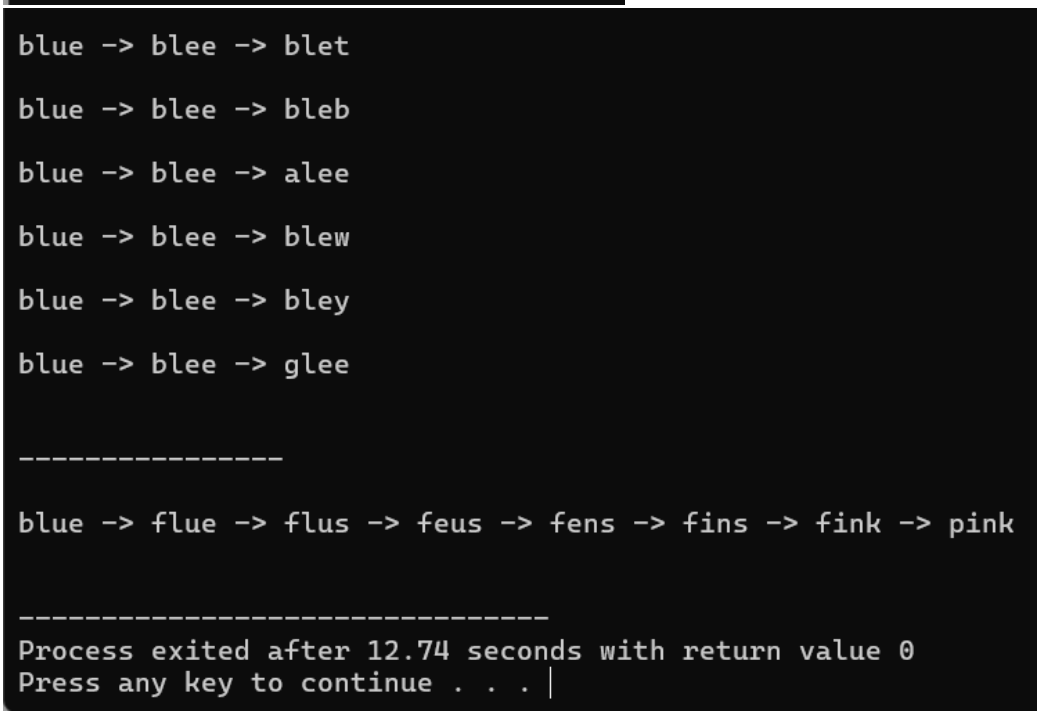
Senaryo 2-

Yalnızca ilk üç adım için queue'yu yazdırdım çünkü çok uzun oluyor, 40-50 sayfa gerekir. Ayrıca her adımda queue'yu yazdırmadığım durumda cevabı 1 saniyenin altında buluyor ancak yazdırırsam 5 dakikayı geçiyor yani hız açısından da maliyeti var.

```
source word: blue

destination word: pink

filename: dictionary.txt


_________________
blue -> flue

blue -> blee

blue -> glue

blue -> slue

blue -> blur

blue -> blae

blue -> clue

blue -> blub


_________________
```

```
_________________
blue -> blee

blue -> glue

blue -> slue

blue -> blur

blue -> blae

blue -> clue

blue -> blub

blue -> flue -> flub

blue -> flue -> flee

blue -> flue -> flux

blue -> flue -> flus

blue -> flue -> floe


_________________
```


----
blue -> glue
blue -> slue
blue -> blur
blue -> blae
blue -> clue
blue -> blub
blue -> flue -> flub
blue -> flue -> flee
blue -> flue -> flux
blue -> flue -> flus
blue -> flue -> floe
blue -> blee -> bled
blue -> blee -> bree
blue -> blee -> slee


blue -> blee -> blet
blue -> blee -> bleb
blue -> blee -> alee
blue -> blee -> blew
blue -> blee -> bley
blue -> blee -> glee
----
blue -> flue -> flus -> feus -> fens -> fins -> fink -> pink

Process exited after 12.74 seconds with return value 0
Press any key to continue . . .

Senaryo 3-

```
source word: bluw

destination word: pink

filename: dictionary.txt

source word bluw is not in the dictionary...
--------------------------------
Process exited after 6.709 seconds with return value 0
Press any key to continue . . .
```

Senaryo 4-

Yine sadece ilk üç iterasyon için queue'yu yazdırdım.

```
source word: devil

destination word: angel

filename: dictionary.txt


----------------
devil -> devel

devil -> kevil


---------------



----------------
devil -> kevil

devil -> devel -> debel

devil -> devel -> revel

devil -> devel -> bevel

devil -> devel -> nevel

devil -> devel -> level
```

```
devil -> devel -> nevel

devil -> devel -> level

devil -> devel -> kevel


----------------



----------------
devil -> devel -> debel

devil -> devel -> revel

devil -> devel -> bevel

devil -> devel -> nevel

devil -> devel -> level

devil -> devel -> kevel


----------------

devil -> devel -> level -> lever -> leger -> luger -> auger -> anger -> angel


--------------------------------
Process exited after 9.498 seconds with return value 0
Press any key to continue . . .
```

Senaryo 5-

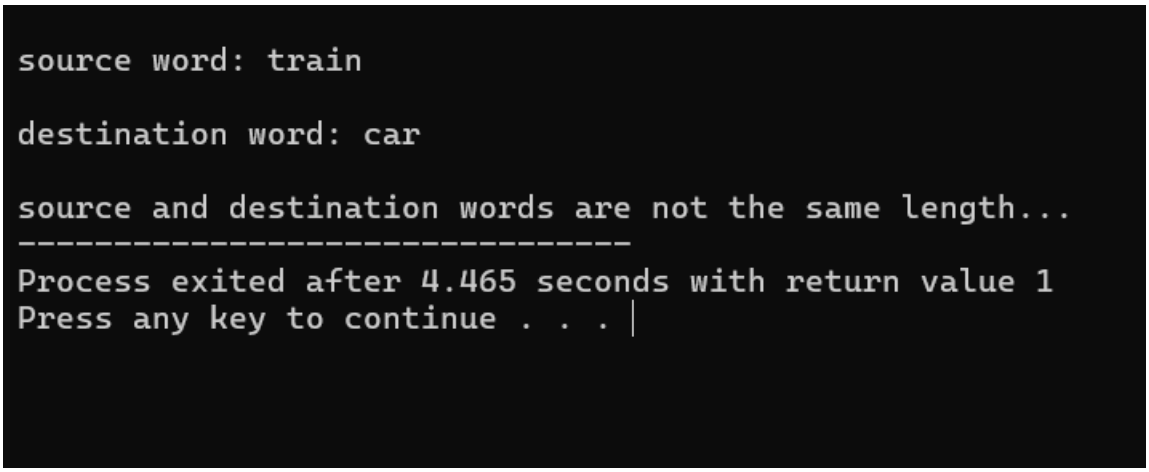
```
source word: heart

destination word: heart

filename: dictionary.txt


----------------

----------------

heart


--------------------------------
Process exited after 10.16 seconds with return value 0
Press any key to continue . . .
```

Senaryo 6-

Yine sadece ilk üç iterasyon için queue'yu yazdırdım.


```
babies -> rabies -> ramies

babies -> rabies -> rubies


----------------


----------------
babies -> babied

babies -> gabies

babies -> rabies -> ramies

babies -> rabies -> rubies


----------------

babies -> rabies -> ramies -> ramees -> rakees -> rakers -> sakers -> sayers -> shyers -> sheers -> steers -> steeps ->
steepy -> sleepy


--------------------------------
Process exited after 11.22 seconds with return value 0
Press any key to continue . . .
```

Senaryo 7-


source word: train
destination word: car
source and destination words are not the same length...
Process exited after 4.465 seconds with return value 1
Press any key to continue . . .