

# Algoritma ve Programlama I

## *Kilit Oyunu*



Siyam Acet 05200000080

# İçindekiler

Programcı kataloğu.....	3
Kullanılan veri yapıları.....	3
Kullanılan fonksiyonlar ve işlevleri.....	3
Test Kataloğu.....	7

# Kullanılan Veri Yapıları



## Liste veri yapısının kullanıldığı yerler

```
EN_FAZLA_YATAY_CIZGI = 8  
HARFLER = list(string.ascii_uppercase[:EN_FAZLA_YATAY_CIZGI + 1])
```

Matriste kullanacağımız harfleri tutmak için string kütüphanesini kullanarak en fazla 8 uzunluklu bir harfler listemiz var.

```
oyun_matris = [[OYUNCU_YOK_TAS for i in range(satir_sutun)] for j in range(satir_sutun)]
```

Oyun matrisini oluştururken iç içe liste kullanıyorum.

## Sözlük veri yapısının kullanıldığı yerler

```
" ")  
mevcut_konum = konuma_cevir(konum[0], oyun_matris)  
hedef_konum = konuma_cevir(konum[1], oyun_matris)
```

Taşın şu anki konumunu ve hedef konumunu kullanıcıdan alırken bunu sözlük veri yapısında saklıyorum.

## Kullanılan Fonksiyonlar ve İşlevleri

```
def satir_sutun_hesapla(tahta_matris):  
    satir_sayi = len(tahta_matris)  
    sutun_sayi = len(tahta_matris[0])  
    return satir_sayi, sutun_sayi
```

Tahtanın satir sutun sayısını hesaplar

```
def tahta_ciz(oyun_matris):  
    satir_sayisi, sutun_sayisi = satir_sutun_hesapla(oyun_matris)  
    print(" " * 6, end="")  
    for harf in HARFLER[:sutun_sayisi]:  
        print(harf, end="")  
        print(" " * 6, end="")  
    print()  
    for satir in range(satir_sayisi):  
        print(" ", TABLO_SATIR_CIZGI * satir_sayisi)  
        print(satir + 1, TABLO_SUTUN_CIZGI, end="")  
  
        for sutun in range(sutun_sayisi):  
            print(" ", oyun_matris[satir][sutun], " ", TABLO_SUTUN_CIZGI, end="")  
  
        print(" ", satir + 1)  
  
    print(" ", TABLO_SATIR_CIZGI * satir_sayisi)  
    print(" " * 6, end="")  
    for harf in HARFLER[:sutun_sayisi]:  
        print(harf, end="")  
        print(" " * 6, end="")  
    print()
```

Tahta\_ciz fonksiyonu her çağrıldığında oyun tahtasını ekrana yazar.

```
def tas_yerlestir(oyun_matris, oyuncu1tas, oyuncu2tas):
    for i in range(len(oyun_matris[0])):
        for j in range(len(oyun_matris[0])):
            if i == 0:
                oyun_matris[i][j] = oyuncu1tas
            if i == len(oyun_matris[0]) - 1:
                oyun_matris[i][j] = oyuncu2tas
```

Oyun başlarken seçilen harflere göre oyun tahtasının ilk ve son yatay çizgisine oyuncuları yerleştirir.

```
def konuma_cevir(harf_kodu, oyun_matris): #girilen konumu sayı olan konuma çevirme
    satir_sayisi, sutun_sayisi = satir_sutun_hesapla(oyun_matris)
    try:
        satir_no = int(harf_kodu[0]) - 1
        sutun_no = HARFLER.index(harf_kodu[1])
        if 0 <= satir_no <= satir_sayisi and 0 <= sutun_no <= sutun_sayisi and len(harf_kodu) == 2:
            return satir_no, sutun_no
    except(ValueError, TypeError, IndexError):
        print("Lütfen geçerli bir konum giriniz! ")
```

Girilen konum değerlerini satır ve sütun numarasına çevirir.

```
def hareket_konum_al_tas_ekle(oyun_matris, oyuncu): #hareket konumunu alıp taşları hareket ettirme fonksiyonu
    while True:
        try:
            konum = input(f"{oyuncu} için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: ").upper().split(" ")
            mevcut_konum = konuma_cevir(konum[0], oyun_matris)
            hedef_konum = konuma_cevir(konum[1], oyun_matris)

            if oyun_matris[mevcut_konum[0]][mevcut_konum[1]] != oyuncu:
                print("Kendi taşınızı oynataniirsiniz!")
                continue

            if len(mevcut_konum) != 2 or len(hedef_konum) != 2:
                print("Geçersiz konum girişi")
                continue
            else:
                if mevcut_konum[0] == hedef_konum[0]:
                    if mevcut_konum[1] > hedef_konum[1]:
                        for i in range(mevcut_konum[1] - 1, hedef_konum[1] - 1, -1):
                            if oyun_matris[mevcut_konum[0]][i] != OYUNCU_YOK_TAS:
                                print("Yerleştirmek istediğiniz taşın konumu dolu!")
                                break
                            else:
                                oyun_matris[mevcut_konum[0]][mevcut_konum[1]] = OYUNCU_YOK_TAS
                                oyun_matris[hedef_konum[0]][hedef_konum[1]] = oyuncu
                                break
```

```

else:
    for i in range(mevcut_konum[1] + 1, hedef_konum[1] + 1):
        if oyun_matris[mevcut_konum[0]][i] != OYUNCU_YOK_TAS:
            print("Yerleřtirmek istediđiniz tařın konumu dolu!")
            break
        else:
            oyun_matris[mevcut_konum[0]][mevcut_konum[1]] = OYUNCU_YOK_TAS
            oyun_matris[hedef_konum[0]][hedef_konum[1]] = oyuncu
            break

elif mevcut_konum[1] == hedef_konum[1]:

    if mevcut_konum[0] > hedef_konum[0]:
        for i in range(mevcut_konum[0] - 1, hedef_konum[0] - 1, -1):
            if oyun_matris[i][mevcut_konum[1]] != OYUNCU_YOK_TAS:
                print("Yerleřtirmek istediđiniz tařın konumu dolu!")
                break
            else:
                oyun_matris[mevcut_konum[0]][mevcut_konum[1]] = OYUNCU_YOK_TAS
                oyun_matris[hedef_konum[0]][hedef_konum[1]] = oyuncu
                break

    else:
        for i in range(mevcut_konum[0] + 1, hedef_konum[0] + 1):
            if oyun_matris[i][mevcut_konum[1]] != OYUNCU_YOK_TAS:
                print("Yerleřtirmek istediđiniz tařın konumu dolu!")
                break
            else:
                oyun_matris[mevcut_konum[0]][mevcut_konum[1]] = OYUNCU_YOK_TAS
                oyun_matris[hedef_konum[0]][hedef_konum[1]] = oyuncu
                break

    else:
        print("Çapraz hareket yok")

except(IndexError, TypeError, ValueError):
    print("Geçersiz konum giriři!")

return hedef_konum

```

Hareket konum al tař ekle fonksiyonu kullanıcıdan konum alıp tařları yerleřtirir.

```

def tas_sil(oyun_matris, satir_sutun, oyuncu):#taş silme fonksiyonu
    for i in range(satir_sutun):
        for j in range(satir_sutun - 2):
            if oyun_matris[i][j] == oyuncu == oyun_matris[i][j + 2] and oyun_matris[i][j + 1] != oyuncu and oyun_matris[i][j + 1] != OYUNCU_YOK_TAS:
                oyun_matris[i][j + 1] = OYUNCU_YOK_TAS
                print(i + 1, HARFLER[j + 1], " konumundaki taş kitlendi")
                tahta_ciz(oyun_matris)
        for i in range(satir_sutun - 2):
            for j in range(satir_sutun):
                if oyun_matris[i][j] == oyuncu == oyun_matris[i + 2][j] and oyun_matris[i + 1][j] != oyuncu and oyun_matris[i + 1][j] != OYUNCU_YOK_TAS:
                    oyun_matris[i + 1][j] = OYUNCU_YOK_TAS
                    print(i + 1, HARFLER[j + 1], " konumundaki taş kitlendi")
                    tahta_ciz(oyun_matris)
        for i in range(satir_sutun):
            for j in range(satir_sutun):
                if i == 0 and j == 0:
                    if oyun_matris[i][j] != OYUNCU_YOK_TAS and oyun_matris[i][j] != oyuncu and oyun_matris[i][j + 1] == oyuncu and oyun_matris[i + 1][j] == oyuncu:
                        oyun_matris[i][j] = OYUNCU_YOK_TAS
                        print(i + 1, HARFLER[j + 1], " konumundaki taş kitlendi")
                        tahta_ciz(oyun_matris)
                    if i == 0 and j == satir_sutun - 1:
                        if oyun_matris[i][j] != OYUNCU_YOK_TAS and oyun_matris[i][j] != oyuncu and oyun_matris[i][j - 1] == oyuncu and oyun_matris[i + 1][j] == oyuncu:
                            oyun_matris[i][j] = OYUNCU_YOK_TAS
                            print(i+1, HARFLER[j+1], " konumundaki taş kitlendi")
                            tahta_ciz(oyun_matris)
                    if i == satir_sutun - 1 and j == 0:
                        if oyun_matris[i][j] != OYUNCU_YOK_TAS and oyun_matris[i][j] != oyuncu and oyun_matris[i - 1][j] == oyuncu and oyun_matris[i][j + 1] == oyuncu:
                            oyun_matris[i][j] = OYUNCU_YOK_TAS
                            print(i+1, HARFLER[j+1], " konumundaki taş kitlendi")
                            tahta_ciz(oyun_matris)
                    if i == satir_sutun - 1 and j == satir_sutun - 1:
                        if oyun_matris[i][j] != OYUNCU_YOK_TAS and oyun_matris[i][j] != oyuncu and oyun_matris[i][j - 1] == oyuncu and oyun_matris[i - 1][j] == oyuncu:
                            oyun_matris[i][j] = OYUNCU_YOK_TAS
                            print(i+1, HARFLER[j+1], " konumundaki taş kitlendi")
                            tahta_ciz(oyun_matris)

```

Taş\_sil fonksiyonu tüm matrisi gezer gerekli gördüğü durumda rakip oyuncunun taşını siler.

```

def tas_hesapla(oyun_matris,oyuncu,satir_sutun):# rakip oyuncunun taşlarını hesaplama fonksiyonu
    rakip_tas_say = 0
    for i in range(satir_sutun):
        for j in range(satir_sutun):
            if oyun_matris[i][j] != oyuncu and oyun_matris[i][j] != OYUNCU_YOK_TAS:
                rakip_tas_say +=1
    return rakip_tas_say

```

Tas\_hesapla fonksiyonu rakip oyuncunun kaç taşı kaldığını hesaplar.

# Test Kataloğu

```
Lütfen tablonun satır/sutun sayısını girin: 4
Lütfen oyuncu 1 için bir harf girin: X
Lütfen oyuncu 2 için bir harf girin: Y

  A      B      C      D
-----
1 | Y  | Y  | Y  | Y  | 1
-----
2 |   |   |   |   | 2
-----
3 |   |   |   |   | 3
-----
4 | X  | X  | X  | X  | 4
-----
  A      B      C      D
X için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 4a 3a
  A      B      C      D
```

Tablonun satır / sütun sayısını 4 , 1. oyuncu X 2. oyuncu için Y karakterini girdiğimizde 4x4'lük bir oyun tahtası karşımıza çıkar. Tablonun en altında X oyuncusu en üstünde Y oyuncusu olur. X oyuncusu 1. oyuncu olduğu için X oyuncusundan konum girmesi istenir. Oyuncunun yazdığı konuma taş hareket ettirilir.

```
  A      B      C      D
-----
1 | Y  | Y  | Y  | Y  | 1
-----
2 |   |   |   |   | 2
-----
3 | X  |   |   |   | 3
-----
4 |   | X  | X  | X  | 4
-----
  A      B      C      D
Y için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 1a 2a
  A      B      C      D
-----
1 |   | Y  | Y  | Y  | 1
-----
2 | Y  |   |   |   | 2
-----
3 | X  |   |   |   | 3
-----
4 |   | X  | X  | X  | 4
-----
  A      B      C      D
```

X oyuncusu taşı hareket ettirdikten sonra sıra Y oyuncusuna geçer ve Y oyuncusundan konum girmesi istenir. Y oyuncusu konum girdikten sonra Y taşı hareket ettirilir. Sırasıyla oyunculardan taşları hareket ettirilmesi istenir.

```

X için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 3d 3c
  A      B      C      D
-----
1 |      |      | Y  | Y  | 1
-----
2 | Y  |      |      |      | 2
-----
3 | X  | Y  | X  |      | 3
-----
4 |      | X  | X  |      | 4
-----
  A      B      C      D
3 B konumundaki taş kitlendi
  A      B      C      D
-----
1 |      |      | Y  | Y  | 1
-----
2 | Y  |      |      |      | 2
-----
3 | X  |      | X  |      | 3
-----
4 |      | X  | X  |      | 4
-----
  A      B      C      D
Y için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 2a 2b
  A      B      C      D

```

Bu durumda 3B konumundaki taşın kitlendiği ve dışarı alındığı hamleyi görüyoruz. Taş dışarı alındıktan sonra sıra Y oyuncusuna geçer.

```

  A      B      C      D
Y için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 1d 2d
  A      B      C      D
-----
1 |      |      | Y  |      | 1
-----
2 |      | Y  | X  | Y  | 2
-----
3 | X  |      |      |      | 3
-----
4 |      | X  | X  |      | 4
-----
  A      B      C      D
2 C konumundaki taş kitlendi
  A      B      C      D
-----
1 |      |      | Y  |      | 1
-----
2 |      | Y  |      | Y  | 2
-----
3 | X  |      |      |      | 3
-----
4 |      | X  | X  |      | 4
-----
  A      B      C      D
X için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 4c 4d
  A      B      C      D

```

Burada da X taşının kitlendiğini görüyoruz.



```

X için hareket ettirilecek taşın konumunu ve hedef konumu giriniz: 3a 2a
  A      B      C      D
-----
1 |      |      |      | 1
-----
2 | X    | Y    | X    | 2
-----
3 |      |      |      | Y  | 3
-----
4 |      | X    |      |      | 4
-----
  A      B      C      D
2 B konumundaki taş kitlendi
  A      B      C      D
-----
1 |      |      |      | 1
-----
2 | X    |      | X    | 2
-----
3 |      |      |      | Y  | 3
-----
4 |      | X    |      |      | 4
-----
  A      B      C      D

```

Oyunun ileriki aşamalarında X oyuncusunun Y taşını kitlediğini görüyoruz.

```

  A      B      C      D
Oyunu kazanan oyuncu: X
Oyuna devam etmek istiyor musunuz? (E,e,H,h) h

```

Y' nin sadece bir taşı kaldığı için Y oyuncusu oyunu kaybediyor. Oyunu X oyuncusu kazanıyor ve oyuna devam edilip edilmeyeceği kullacıya soruluyor.