YAZILIM SINAMA ÖDEV1

1. AMAÇ

Bir santranç tahtası üzerine santranç piyonlarını ilk piyon siyah olmayacak ve siyah iki piyon yan yana gelmeyecek şekilde kaç farklı dizilim bulunduğunu sayısını ve dizilimini ekrana yazdırmak.

1. GİRDİLER

count 🡪 sayıların hesaplamak için

toplamtas 🡪piyon tasların tamamı

siyahtas 🡪 piyon taslarının siyah olanlar

tahta 🡪 tahta için birli bir liste oluşturma

tahtayeni 🡪 tahta için ikili iç içe bir liste oluşturma

toplam 🡪 countların toplanacağı herolasılığın sonucunun bulunduğu değer

secilentas 🡪 seçilen piyon adedini yazıldığı değer

1. ÇÖZÜM MANTIĞI

Seçilen taşın yarısı kadar değerlerin her siyah piyon adedine göre hesaplaması için pemütasyon fonksiyonuna yönlenerek her olasılığın olasılığını hesaplatıp yazan ve sayısını counta göndererek bu countları toplatıp toplam olasılığı yazdıran bir algoritma mantığı oluşturmak.

1. PROGRAM KODU

toplam,secilentas=0,8

def permutasyon(toplamtas,siyahtas,hesap):

count=0

tassay=toplamtas-siyahtas-hesap

print(tassay)

tahta=[0 for i in range(toplamtas)]

tahtayeni=[[0 for i in range(toplamtas)]for j in range(tassay)]

if siyahtas==0:

for tas in range(0,toplamtas):

tahta[tas]='b'

print(tahta[tas]," ",end="")

count+=1

elif siyahtas==1:

for tas in range(0,toplamtas):

count=0

if tas!=0:

print(tahta[0]," ",end="")

for x in range(1,toplamtas):

if x==tas:

tahta[tas]='s'

else:

tahta[tas]='b'

if tas!=0:

print(tahta[tas]," ",end="")

count+=1

print()

else:

for i in range(len(tahtayeni),0,-1):

for j in range(i):

for tas in range(0,toplamtas):

if tas==0:

tahtayeni[j][tas]='b'

elif tas==1 :

if tassay==i :

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==2:

if tassay-1==i :

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==3:

if tassay-2==i or (tassay==i and j==0):

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==4:

if tassay-3==i or (tassay-1==i and j==0) or (tassay==i and j==1):

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==5:

if tassay-4==i or (tassay-2==i and j==0) or (tassay-1==i and j==1) or(tassay==i and j==2) or (siyahtas>2 and tassay==i and j==0):

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==6:

if tassay-5==i or (tassay-3==i and j==0) or (tassay-2==i and j==1) or (tassay-1==i and j==2) or(tassay==i and j==3) or (siyahtas>2 and tassay-1==i and j==0) or (siyahtas>2 and tassay==i and j==1):

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

elif tas==7:

if tassay-6==i or (tassay-4==i and j==0) or (tassay-3==i and j==1) or (tassay-2==i and j==2) or (tassay-1==i and j==3) or(tassay==i and j==4) or (siyahtas>2 and tassay-1==i and j==1) or (siyahtas>2 and tassay-2==i and j==0) or (siyahtas>2 and tassay==i and j==2) or (siyahtas>3 and tassay==i and j==0):

tahtayeni[j][tas]='s'

else:

tahtayeni[j][tas]='b'

print(tahtayeni[j][tas]," ",end="")

print()

count+=1

return count

for santranc in range (int((secilentas)/2)+1):

count=permutasyon(secilentas,santranc,santranc-1)

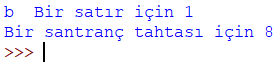
toplam+=count

print ("Bir satır için",toplam)

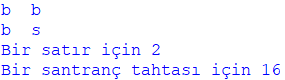
print ("Bir santranç tahtası için",toplam\*8)

1. EKRAN GÖRÜNTÜSÜ

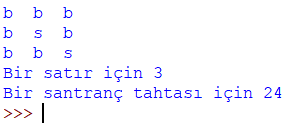
1x1



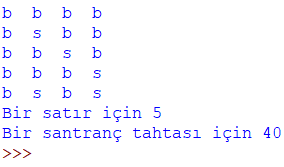
2x2



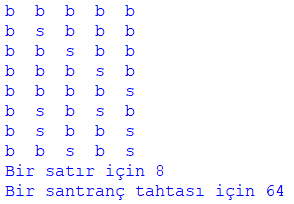
3x3



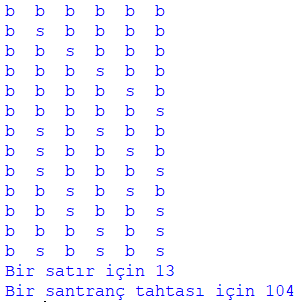
4x4



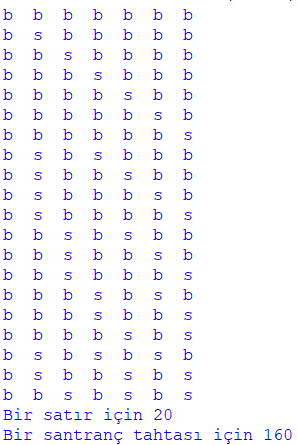
5x5



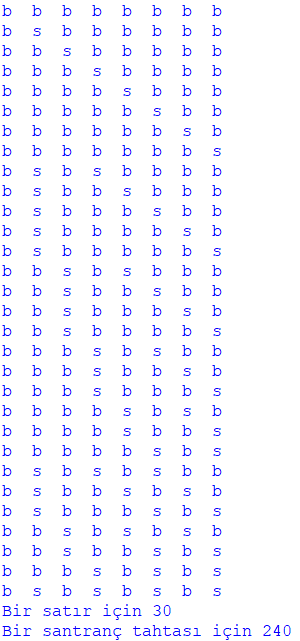
6x6



7x7



8x8



1. TEST CASE
2. TEST CASE 2: Yılanın hareket testi

|  |  |
| --- | --- |
| **RISK LEVEL** | **HIGH** |
| **PURPOSE** | Gerçekleştirilen programın doğru bir şekilde çalıştığının gösterilmesi |
| **INPUTS** | secilentas değişkenine 9 değerini atayarak |
| **EXPECTED OUTPUTS** | Sıralı dizlimin ve olasılık sayısının doğru bir şekide ekrana gelmesi |
| **PASS CRITERIAS** | Sıralı dizilimin düzgün gözükmesi |
| **FAIL CRITERIAS** | Algoritmanın sorunsuz devam etmesi. |
| **TEST PROSEDURE** | Kod bloğunu açtım  Kod bloğumda secilentas değişkenini 9’a değiştirdim  Kod bloğunu çalıştıra basdım |