



HASAN FERDİ TURGUTLU TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ

YZM 3105 - YAZILIM SINAMA
ÖDEV 1 – Beyaz-Siyah Taş Problemi

142805010

ABDULLAH ÇALIŞKAN

142805001

ERHAN AYGÖREN

1. Beyaz-Siyah Taş Problemi

Beyaz-Siyah taşların; girilen basamak sayısı ve belirtilen kurallara göre oluşturabileceği dizilimin kaç tane olduğunun bulunması.

Kurallar:

- Her dizilimin en başındaki taş Beyaz olmalıdır.
- Siyah taşlar yan yana gelmemelidir.

1.1. Amaç

Girilen basamak sayısına göre, varolan dizilim sayısının doğru bir şekilde bulunması ve matematiksel formül kullanmadan, bilgisayarın gücü ile dizilimin kaç tane olduğunun bulunması.

1.2. Girdiler

BasamakSayisi (n)	Integer	Bir dizilimde kullanılacak max. taş sayısını da ifade eder.
-------------------	---------	---

1.3. Mantığı

Kullanıcının girdiği basamak sayısına göre, kaç defa siyah taş yerleştirilmeye çalışılacağını hesaplar.

Bu hesaplama, `calc_limit()` fonksiyonu tarafından yapılır ve bir limit değeri elde edilir.

Daha sonrasında, `s()` fonksiyonu ile koyulabilecek siyah taşların sayısı elde edilir. `S()` fonksiyonu burada recursive olarak çalışmaktadır. Fonksiyon sürekli, boyutu azalan bir dizi alır ve kaç siyah taş yerleştirilebileceğini hesaplamaya çalışır. Fonksiyonun çıkış noktası ise, sayı değerinin yani kaç tane siyah taş konulacağını gösteren sayı değerinin 0 veya 1 olmasına göre sonlanmasıdır.

Örnek:

Kullanıcı basamak sayısı olarak, 4 girerse; $n = 4$

ilk iş, uzunluğu 4 olan bir liste oluşturulur ve varsayılan değer olarak 1 atanır. (1:Beyaz, 0:Siyah)

Kaç kez siyah taş yerleştirmeye çalışılması gerektiği hesaplanır. `calc_limit(4)`, `limit = 3`

Ve for döngüsüne geldik.

- $L=0$, Dizinin ilk elemanı Beyaz taş yapılıyor ve 4 uzunluğundaki bir listenin, ilk indisi hariç diğer 3 değeri ile yeni bir liste oluşturuluyor. Bu liste, `s(0, new_main)` fonksiyonuna gönderiliyor. 0 olmasının sebebi, Hiç siyah taş olmama durumunun temsil edilmesidir ve toplam değerimiz 1 olur. Toplam = 1
- $L=1$, Dizinin ilk elemanı Beyaz taş yapılıyor ve 4 uzunluğundaki bir listenin, ilk indisi hariç diğer 3 değeri ile yeni bir liste oluşturuluyor. Bu liste, `s(1, new_main)` fonksiyonuna gönderiliyor. Şuanda bu diziye 1 tane siyah taş kaç farklı şekilde yerleştirilebilir sorusunun cevabının temsili. Fonksiyon burada dizinin uzunluğu kadar bir değer döndürür. Dönüş değeri 3 olur ve yeni toplam = 4 olur.

- L=2, Dizinin ilk elemanı beyaz ve 3 uzunluğunda yeni bir liste oluşturuluyor. Bu liste, s(2, new_main) fonksiyonuna gönderiliyor. Fonksiyon, sayının 2 olduğunu görüyor ve ilk koyacağı Siyah taşı kaç kere kaydıracağını hesaplıyor. Yani SB- olduğunda dizi, acaba bir kere daha kaydıracak mıyım? Şu an 3 lük bir dizi olduğu için sonuç 1 kere dönmek olur. Eğer ki new_main 4 lük bir dizi olsaydı olabilecek dizilim, SBSB SBBS BSBS olabileceği için siyah taş kaydırılması gerekir. Bu hesaplama işlemi dizinin uzunluğundan sayının çıkarılmasıyla bulunur. Daha sonra 3 lük dizinin içerisinde SB- şekline getiriyor ve kalan son 1 basamağı tekrar kendisine yolluyor. S(1, new_main) fonksiyon tekrar çalıştığında sayının 1 değeri için, dizinin uzunluğunu dönüyor. Yani 1 değeri. Geri dönüş değeri 1 oluyor. Siyah taş kaydırılmayacağı için, s fonksiyonu içindeki for döngüsü sona eriyor ve geri dönüş değeri 1 oluyor. Yeni toplam = 5 oldu.

1.4. Program Kodu : Python3.x – Linux 4.4.2

```
# -*- coding: utf-8 -*-
# /usr/bin/env python3
import math

def s(sayi, dizi):
    if sayi == 0:
        return 1
    elif sayi == 1:
        return len(dizi)
    else:
        limit = len(dizi) - sayi
        i = 0
        toplam = 0
        for l in range(limit):
            dizi[i] = 0
            dizi[i+1] = 1
            new_dizi = dizi[i+2:]
            toplam += s(sayi-1, new_dizi)
            i += 1
        return toplam

def calc_limit(n):
    if n % 2 == 0:
        return int((n / 2) + 1)
    else:
        return int(math.ceil(n / 2))

if __name__ == '__main__':
    try:
        n = int(input("Number :"))
    except:
        raise TypeError("Number değeri, integer olmalıdır.")
    if n <= 0:
        print("N değeri 0 veya daha küçük bir değer olamaz.")
        return

    main = []
    for i in range(n):
        main.append(1)
    limit = calc_limit(n)

    toplam = 0
    for l in range(limit):
        main[0] = 1
        new_main = main[1:]
        toplam += s(l, new_main)
    print(toplam)
```

1.5. Ekran Çıktısı

```
acaliskan@syndrome:~/learning/Python/yazilim_sinama_odev$ python3 main.py
Number :test
Traceback (most recent call last):
  File "main.py", line 32, in <module>
    n = int(input("Number :"))
ValueError: invalid literal for int() with base 10: 'test'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "main.py", line 34, in <module>
    raise TypeError("Number değeri, integer olmalıdır.")
TypeError: Number değeri, integer olmalıdır.
acaliskan@syndrome:~/learning/Python/yazilim_sinama_odev$ python3 main.py
Number :-1
Traceback (most recent call last):
  File "main.py", line 36, in <module>
    raise ArithmeticError("N değeri 0 veya daha küçük bir değer olamaz.")
ArithmeticError: N değeri 0 veya daha küçük bir değer olamaz.
acaliskan@syndrome:~/learning/Python/yazilim_sinama_odev$ python3 main.py
Number :15
987
acaliskan@syndrome:~/learning/Python/yazilim_sinama_odev$ python3 main.py
Number :20
10946
acaliskan@syndrome:~/learning/Python/yazilim_sinama_odev$ python3 main.py
Number :30
1346269
```

1.6. Test Case

a. Programın doğru sonuç ürettiğinin kontrol edilmesi

Risk	Yüksek
Amaç	Problemin çözümünün doğruluğunun tespit edilmesi.
Girdiler	Basamak değeri. (3,4,15,20)
Beklenen Çıktılar	Basamak değerine göre, doğru değer oluşması. (3, 5, 987, 10946)
Test Geçiş Kriteri	Beklenen çıktının elde edilmesi.
Başarısızlık Kriteri	Basamak değerine göre hesaplama sonucunda, toplamın beklenen çıktıya denk olmaması.
Test Prosedürü	Test kullanıcısı; yazılımı, desteklenen sistem ve gerekliliklerini karşılayarak testi gerçekleştirmelidir. (Python3.x ve Linux 4.4.2) Test işlemini tamamladıktan sonra test sonucunu, Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır.

b. Basamak değerinin negatif girilmesinin testi

Risk	Yüksek
Amaç	Negatif bir basamak girdisi ile programın exception üretmesinin test edilmesi
Girdiler	Basamak değeri. (-1, -19, -100)
Beklenen Çıktılar	ArithmeticError()
Test Geçiş Kriteri	ArithmeticError exceptionının elde edilmesi
Başarısızlık Kriteri	Programın exception üretmeme durumu
Test Prosedürü	Test kullanıcısı; yazılımı, desteklenen sistem ve gerekliliklerini karşılayarak

	testi gerçekleştirmelidir. (Python3.x ve Linux 4.4.2) Test işlemini tamamladıktan sonra test sonucunu, Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır.
--	--

c. Basamak değerinin integer dışında bir veri tipi olarak girilmesi

Risk	Yüksek
Amaç	Integer olmayan bir veri girdisinde, exception üretmesinin tespit edilmesi
Girdiler	Basamak değeri (test, abc)
Beklenen Çıktılar	TypeError()
Test Geçiş Kriteri	TypeError exceptionın fırlatılmış olması
Başarısızlık Kriteri	TypeError exceptionın fırlatılmamış olması
Test Prosedürü	Test kullanıcısı; yazılımı, desteklenen sistem ve gerekliliklerini karşılayarak testi gerçekleştirmelidir. (Python3.x ve Linux 4.4.2) Test işlemini tamamladıktan sonra test sonucunu, Pass/Fail olarak belirterek nedenleri ile birlikte raporlamalıdır.