

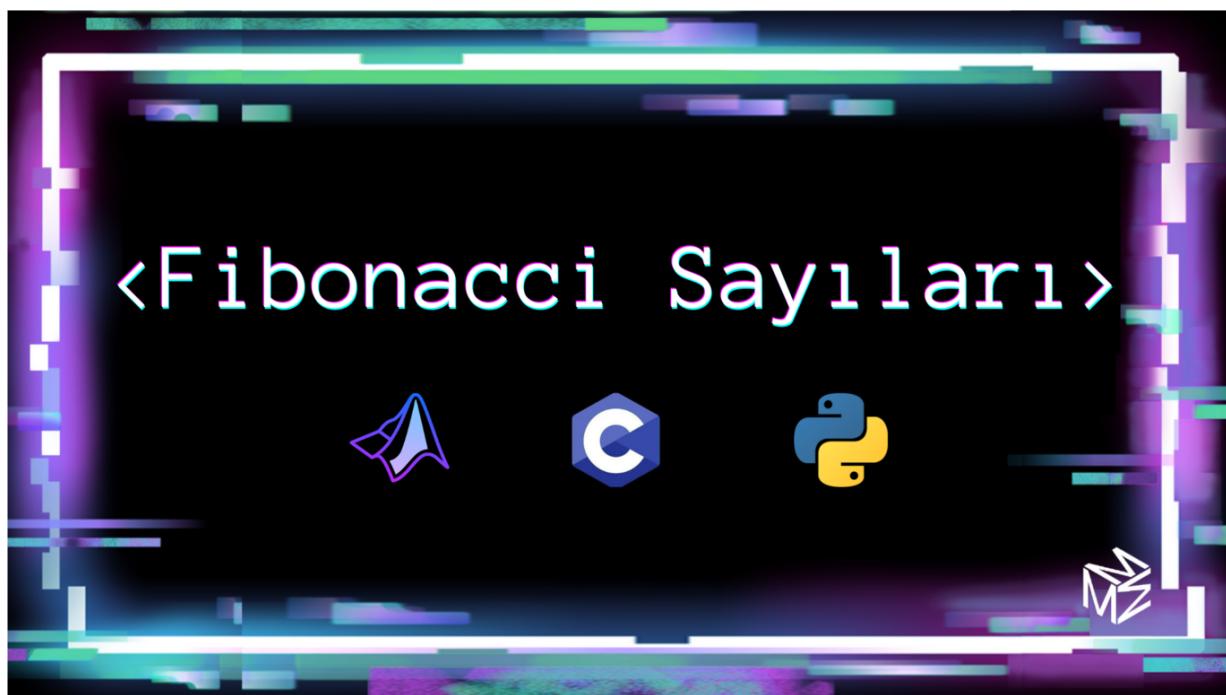
#Terminal_001: Fibonacci Sayıları

Matematik ve Matematik Mühendisliği Topluluğu¹

¹Affiliation not available

December 7, 2020

Bu haftaki yazımızda, ünlü matematikçi Fibonacci'nin varsayımsal bir tavşan popülasyonunu modelleyerek elde ettiği Fibonacci dizisini C, Python ve MATLAB'de kodlayarak inceledik.



Aşağıdaki numaralardan yazı içerisinde kolayca ilerleyebilirsiniz.

- 1. Fibonacci Sayıları ve Scrum**
- 2. Fibonacci Sayıları Nereden Gelir?**
- 3. Fibonacci Sayılarının C ile Bulunması**
- 4. Fibonacci Sayılarının Python ile Bulunması**
- 5. Fibonacci Sayılarının MATLAB ile Bulunması**

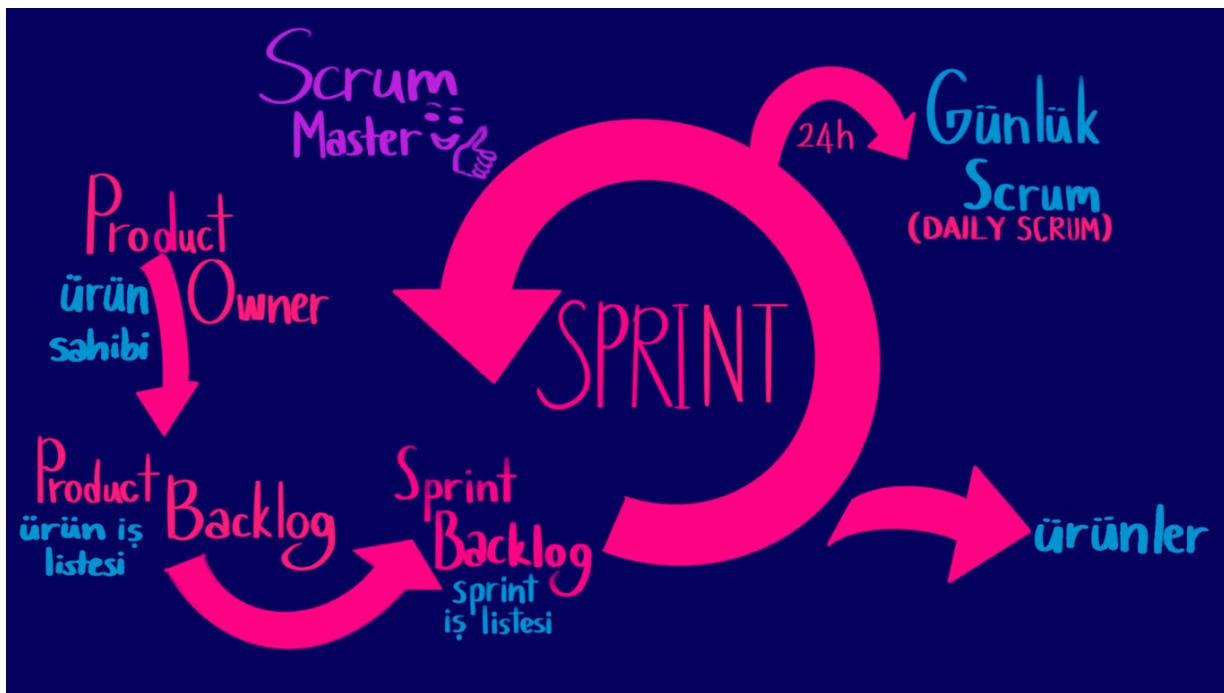


Figure 1: Fibonacci Sayıları ve Scrum

MMMT olarak topluluk içinde Agile¹ altında Scrum² çerçevesi ile çalışıyoruz. Bu çerçevede her Sprint³ Planlama toplantımızda o Sprint'te yapacağımız işlerin ne kadar efor gerektireceğini göreceli tahminleme/eforlama yaparak belirlemek için Fibonacci sayılarını kullanıyoruz. Bu sebeple MMMT üyelerinin hayatında Fibonacci sayıları çok önemli bir yer kaplıyor! Siz günlük hayatınızda Fibonacci sayılarına rastlıyor musunuz?

1. *Agile: Çevik, belirsiz bir ortamda dahi başarılı olabilmek için değişime karşılık verebilme ve geliştirme esnekliği, kapasitesi ve yeteneğidir.*
2. *Scrum: İnsanların mümkün olan en yüksek değere sahip ürünler üretken ve yaratıcı bir şekilde geliştirirken, karmaşık ve adaptasyona açık soruları çözer.*
3. *Sprint: İçerisinde "Bitti" durumunda, kullanılabilir ve potansiyel olarak yayınlanabilir bir Ürün Parçasının oluşturduğu, bir ay veya daha az vaka süresi.*



Figure 2: **Fibonacci Sayıları Nereden Gelir?**

Fibonacci sayıları; $1, 1, 2, 3, 5, 8, 13, 21, \dots$ sayı dizisinin elemanları olup her biri kendinden önceki iki sayının toplamıdır. Bu sayılar ilk olarak Orta Çağ İtalyan matematikçisi Leonardo Pisano (Fibonacci) tarafından Avrupa'da kayda geçirilmiş, Hindu-Arap rakamlarını ve ondalık sayı sistemini popüler hale getiren Liber Abaci'sinde (1202; "Abaküs Kitabı") not edilmiştir. (Britannica, 2020).

Fibonacci aşağıdaki durumları modellemiştir;

- Yeni doğmuş tavşan çifti (bir dişi, bir erkek) bir ağıla⁴ konur.
- Tavşanlar bir aylıkken çiftleşir.
- Tavşanların bir aylık gebelik süresi vardır.
- Tavşanların asla ölmediği ve bu dişinin ikinci aydan itibaren her ay yeni bir çift (bir erkek, bir dişi) doğurduğu varsayılr.
- Bir yılın sonunda kaç tane dişi tavşan vardır?

4. Ağıl: Evcil küçükbaş hayvanların barındığı çit veya duvarla çevrili yer, arkaç.

Burada dişi tavşan sayısı, erkek tavşan sayısı ve tavşan çifti sayısı aynı sayıya tekabül eder.

- 1- Birinci ayın sonunda bir dişi olacaktır.
- 2- İkinci ayın sonunda hala bir dişi olacaktır ancak dişi hamiledir.
- 3- Üçüncü ayın sonunda iki dişi olacaktır: biri hamiledir ve diğerinin hamile değildir.

Genel olarak, dişiler (n) = dişi ($n-1$) + dişi ($n-2$) ile ifade edilir.

- $n-2$ 'de yaşayan her dişi n ayında bir dişi doğuracaktır;
- $n-2$ 'de yaşayan bu dişiler n . aydaki toplam dişileri bulmak için $n-1$ 'de yaşayanlara eklenebilir.

Ay	Diş Sayısı
1	1
2	1
3	2
4	3
5	5
6	8



Temel durumlar:

$$* \text{ Dişiler (1)} = 1$$

$$* \text{ Dişiler (2)} = 1$$

Özyinelemeli⁵ durum:

$$* \text{ Dişiler (n)} = \text{Dişiler (n-1)} + \text{Dişiler (n-2)}$$

Neden Özyinelemeli durumda sondan önceki bir ay ve sondan önceki iki ayda bulunan tavşan sayısını topluyoruz? Çünkü iki ay önceki tavşan çifti sayısı kadar yeni tavşan doğacak ve bir önceki tavşan çifti sayısı kadar tavşan yaşamaya devam edecek.

Peki, bir önceki ayda bulunan toplam tavşan çifti sayısını nasıl bulacağız? Aynı işlemi tekrar etmemiz gereklidir. İşte bu noktada recursion bize çok kolaylık sağlıyor. Koşulumuz sağlanana kadar fonksiyonumuz kendini çağırarak bizim yerimize bu işlemi getirecektir. Neticede aradığımız aydan bir ve iki ay önce bulunan tavşan çifti sayısını topladık. Nitekim aranan ayda bulunan toplam tavşan sayısını elde etmiş olduk.

⁵. Özyineleme: Genel anlamıyla bir yapının yinelenmesidir. Recursion olarak bilinir.



Figure 3: Fibonacci Sayılarının C İle Bulunması

```
#include <stdio.h>
/* \selectlanguage{ngerman}Öncelikle temel fonksiyonlar\selectlanguage{polish}ı \selectlanguage{english}
ç\selectlanguage{polish}ı\selectlanguage{english}kt\selectlanguage{polish}ı \selectlanguage{english}i\sele
/* Kullanaca\selectlanguage{polish}ğı\selectlanguage{english}m\selectlanguage{polish}ı\selectlanguage{en
int aySayisi, tavsanciftiSayisi;
/* Girilen ay de\selectlanguage{polish}ğ\selectlanguage{english}erine kar\selectlanguage{polish}şı\sele
fonksiyon tan\selectlanguage{polish}ı\selectlanguage{english}mlayal\selectlanguage{polish}ı\selectlanguage{en
int fibonacci(int\selectlanguage{polish} aySayisi){
    /* Eğer aySayisi 1 veya 2 ise direkt olarak 1 sayısını döndürelim. */
    if (aySayisi == 1 || aySayisi == 2)
        return 1;
    /* Değilse sondan bir önceki ay ve sondan iki önceki aydaki tavşan çifti sayısını toplayıp çıkan de
    döndürelim. */
    else
        return fibonacci(aySayisi-1) + fibonacci(aySayisi-2);
}\selectlanguage{ngerman}

/* Ana fonksiyonumuzu tan\selectlanguage{polish}ı\selectlanguage{english}ml\selectlanguage{polish}ı\sele
int main()\selectlanguage{polish} {
    /* Kullanıcının hesaplanacak değeri girmesi için ekrana bir talep cümlesi yazdıralım. */
    printf("Lütfen tavşan çifti sayısını bulmak istediğiniz ayı tam sayı cinsinden giriniz: ");
    /* Hesaplanacak değeri kullanıcından alıp "aySayisi" değişkenine atayalım. */
    scanf("%d",&aySayisi);
    /* Kullanıcıdan alınan değeri "fibonacci()" fonksiyonuna göndererek girilen ay değerinde kaç tavşan
    tavsanciftiSayisi = fibonacci(aySayisi);
```

```

/* Bulduğumuz değeri ekrana yazdıralım. */
printf("Tavsan sayısı= %d \n",tavsanCiftiSayisi);
/* Programın hatasız sonlandırıldığını belirtmek için de "0" döndürelim. */
return 0;
}\selectlanguage{ngerman}

```

Programın çıktısı:

```

Lütfen tavşan çifti sayısını bulmak istediğiniz ayı tam sayı cinsinden giriniz: 10
Tavsan sayısı= 55
Program ended with exit code: 0

```



Figure 4: Fibonacci Sayılarının Python İle Bulunması

```

def fibonacci(aySayisi):
    #E\selectlanguage{polish}\selectlanguage{english}er arad\selectlanguage{polish}ığı\selectlanguage{english}
    if aySayisi <= 2:
        return 1
    else:
        return fibonacci(aySayisi-1) + fibonacci(aySayisi-2)

#Kullan\selectlanguage{polish}ı\selectlanguage{english}c\selectlanguage{polish}ı\selectlanguage{english}
#Ald\selectlanguage{polish}ığı\selectlanguage{english}m\selectlanguage{polish}ı\selectlanguage{english}
aySayisi = int(input("L\selectlanguage{ngerman}ütfen tav\selectlanguage{polish}ş\selectlanguage{english}")

#Yazd\selectlanguage{polish}ığı\selectlanguage{english}m\selectlanguage{polish}ı\selectlanguage{english}
print("Tav\selectlanguage{polish}ş\selectlanguage{english}an Say\selectlanguage{polish}ı\selectlanguage{english}")

```

```
# Burada d\selectlanguage{ngerman}öndürdü\selectlanguage{polish}\selectlanguage{ngerman}ümüz cevab\sele  
#Dilersek \selectlanguage{ngerman}örne\selectlanguage{polish}\selectlanguage{english}in tav\selectlang  
#tavsanSayisi=fibonacci(aySayisi)  
#print(tavsanSayisi)
```

Programın çıktısı:

```
Lütfen tavşan çifti sayısını bulmak istediğiniz ayı tam sayı cinsinden giriniz: 10  
Tavşan Sayısı: 55
```



Figure 5: Fibonacci Sayılarının MATLAB İle Bulunması

```
n=input('L\selectlanguage{ngerman}ütfen tav\selectlanguage{polish}\selectlanguage{english}an \selectlan  
%Kullanıcının hesaplanacak değeri girmesi için ekrana bir talep cümlesi yazdıralım.  
%Hesaplanacak değeri kullanıcından alıp 1x1 bir n matrisine atayalım  
  
%Hesaplanan her ayda kaç çift tavşan bulunduğu bilgisini bir a matrisinde saklayalım.  
a(1)=1;  
a(2)=1;  
%Birinci ve ikinci aydaki tavşan sayısı için direkt olarak 1 sayısını döndürelim.  
  
%Girilen ay değerine karşılık kaç tavşan çifti olduğunu hesaplayacağımız bir fonksiyon tanımlayalım.  
%Döngünün üçüncü aydan itibaren girilen değere kadar birer birer artacak şekilde tüm ayları  
%hesaplaması için ":" kullanalım. Matrisin ikiden sonraki tüm  
%elemanları kendisinden iki önceki elemanın toplamı şeklinde hesaplansın.  
for i=3:n
```

```

    a(i)=a(i-1)+a(i-2);
end

%Son olarak fprintf fonksiyonlarını kullanarak bulduğumuz bu değerleri ekrana yazdıralım.

fprintf("%d. aydaki tavşan sayısı: ",n);
if(n==1)
    i=1;
elseif (n==2)
    i=1;
end
fprintf("%d\n",a(i));

```

Programın çıktısı:

```

>> fibonacci
Lütfen tavşan çifti sayısını bulmak istediğiniz ayı tam sayı cinsinden giriniz:10
10. aydaki tavşan sayısı: 55
>>

```

MATLAB, bilimsel hesaplamalar için çok sık kullanılan bir araç olduğundan bazı özel matematiksel kavramlar için hazır fonksiyonlar bulundurur. Örneğin fibonacci (n) fonksiyonu Fibonacci serisinin n. terimini hesaplar. Yukarıda açıkladığımız algoritmayı bu fonksiyonu kullanarak tekrar yazalım:

```

n=input('serinin kaçinci sayısının hesaplanması istersiniz:');
a=fibonacci(n);
fprintf("serinin %d.sayısı:%d\n",n,a);

```

Programın çıktısı:

```

>> fibonacci_function
serinin kaçinci sayısının hesaplanması istersiniz:10
serinin 10.sayısı:55
>>

```

Figure 6:

Bu haftaki yazımızda, Fibonacci dizisini C, Python ve MATLAB’de kodlayarak inceledik ve bu dizideki sayıların scrumda nasıl kullanıldığına değindik. Önümüzdeki terminal içeriklerinde görüşmek üzere. **Yorumlarınızı sosyal medya hesaplarınızdan bizlerle paylaşmayı unutmayın!**

Instagram: [@mmmt_digital](#)

LinkedIn: [@mmmt-digital](#)

Twitter: [@mmmt_digital](#)

Yazımıza atıfta bulunmak için aşağıdaki gibi APA formatını kullanabilirsiniz:

MMMT. (2020, Aralık 4). #Terminal 001: Fibonacci Sayıları. -erişim tarihi- tarihinde, www.authorea.com/496997/QrdZQSsTz-4FAye6LnK9Gg adresinden erişildi.

Referanslar

- [1] Britannica. (2020, Nisan 12). *Fibonacci numbers*. 11 Kasım 2020 tarihinde, www.britannica.com/science/Fibonacci-number adresinden erişildi.
- [2] edX. (t.y.). *MITx: 6.00.1x: Introduction to Computer Science and Programming Using Python*. 11 Kasım 2020 tarihinde, courses.edx.org/courses/course-v1:MITx+6.00.1x+2T2020a/course/ adresinden erişildi.
- [3] MathWorks. (t.y.) *Fibonacci*. 11 Kasım 2020 tarihinde, www.mathworks.com/help/symbolic/fibonacci.html adresinden erişildi.