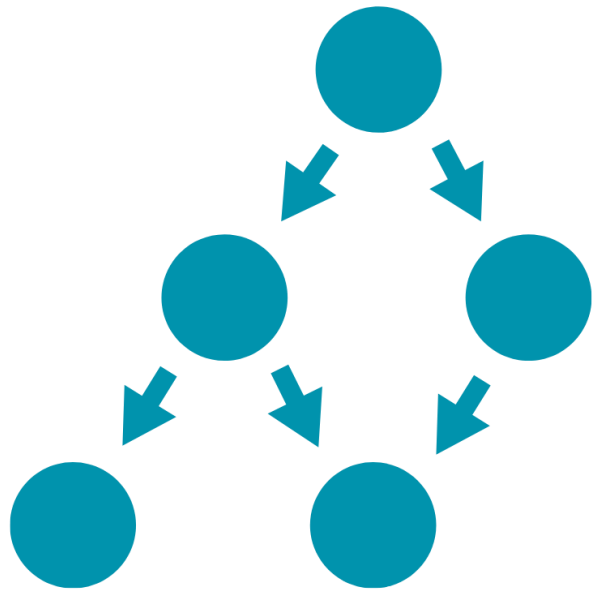


VERİ YAPILARILARI VE ALGORİTMALAR

Stack

«A data structure is a particular way of organizing data in a computer so that it can be used **efficiently***.»*

Giriş

- Yığın Kavramı
- Yığınlar nasıl kullanılır?
- Yığın soyut veri türü
- Yığın uygulamaları
- Performans ve Sınırlar
- Uygulama
  - Dizi ile yığın tasarımı
  - Bağlı liste ile yığın tasarımı

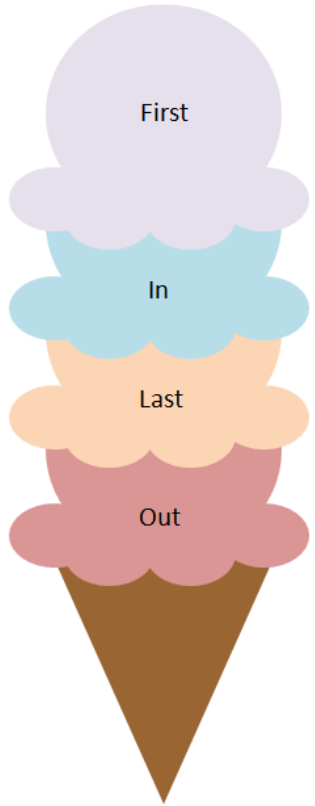
Yığın (Stack)

- Veriler doğal olarak listeler halinde düzenlenir.
- Array ve ArrayList yapısı verileri liste düzeninde organize etmek üzere kullanılan yapılar arasındadır.
- Anlaşılması kolay soyutlamalar sağlayan liste yapılarından biri yığınlardır.

Yığın (Stack)

- Yığın yapısında listeye ekleme işlemi sona yapılır ve listeden çıkarma işlemi de yine son eleman dikkate alınarak gerçekleştirilir.

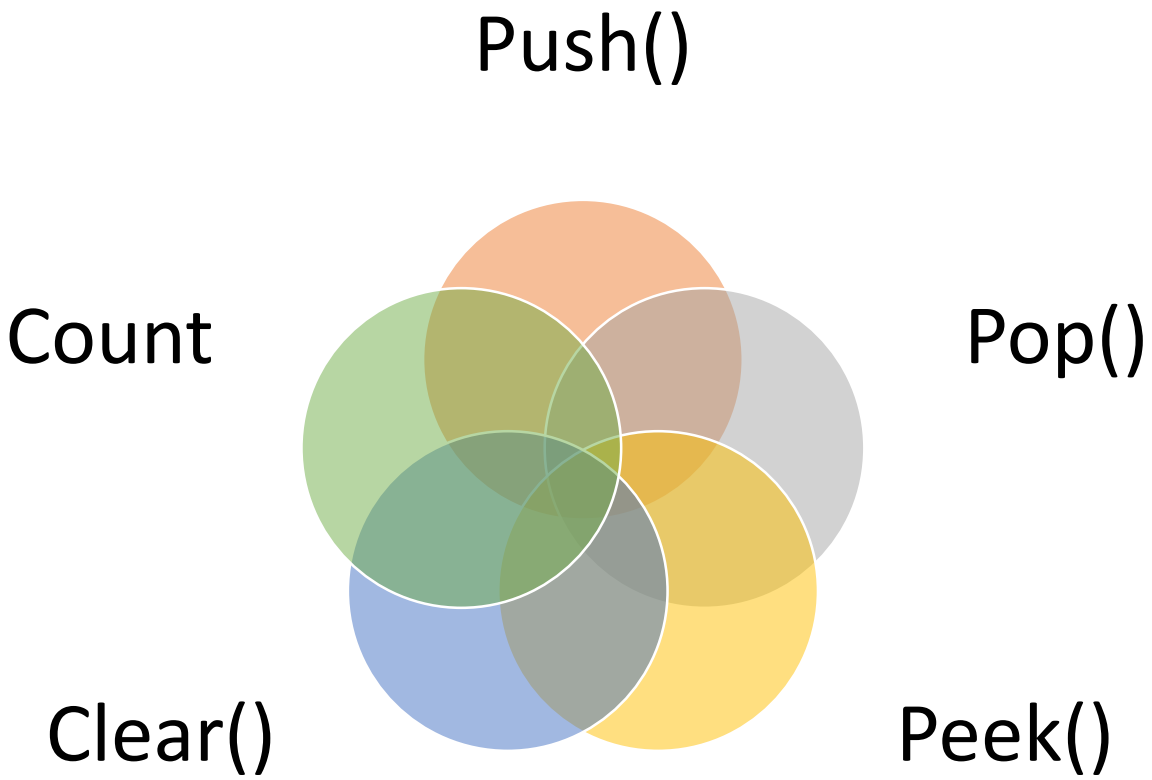
- Yığınlar ifadelerin değerlendirilmelerinden, işlev çağrılarına kadar bilgisayar bilimlerde yaygın bir şekilde kullanılır.



Yığın (Stack)

- Yığınların sadece son elemanlarına erişim sağlanabilir.
- Bu nedenle son-giren ilk-çıkar (last-in first-out, **LIFO**) veri yapısı olarak tanımlanır.
- C# dilinde Stack veri yapısı hem object hem de Generic olarak koleksiyonlar kapsamında sunulmaktadır.

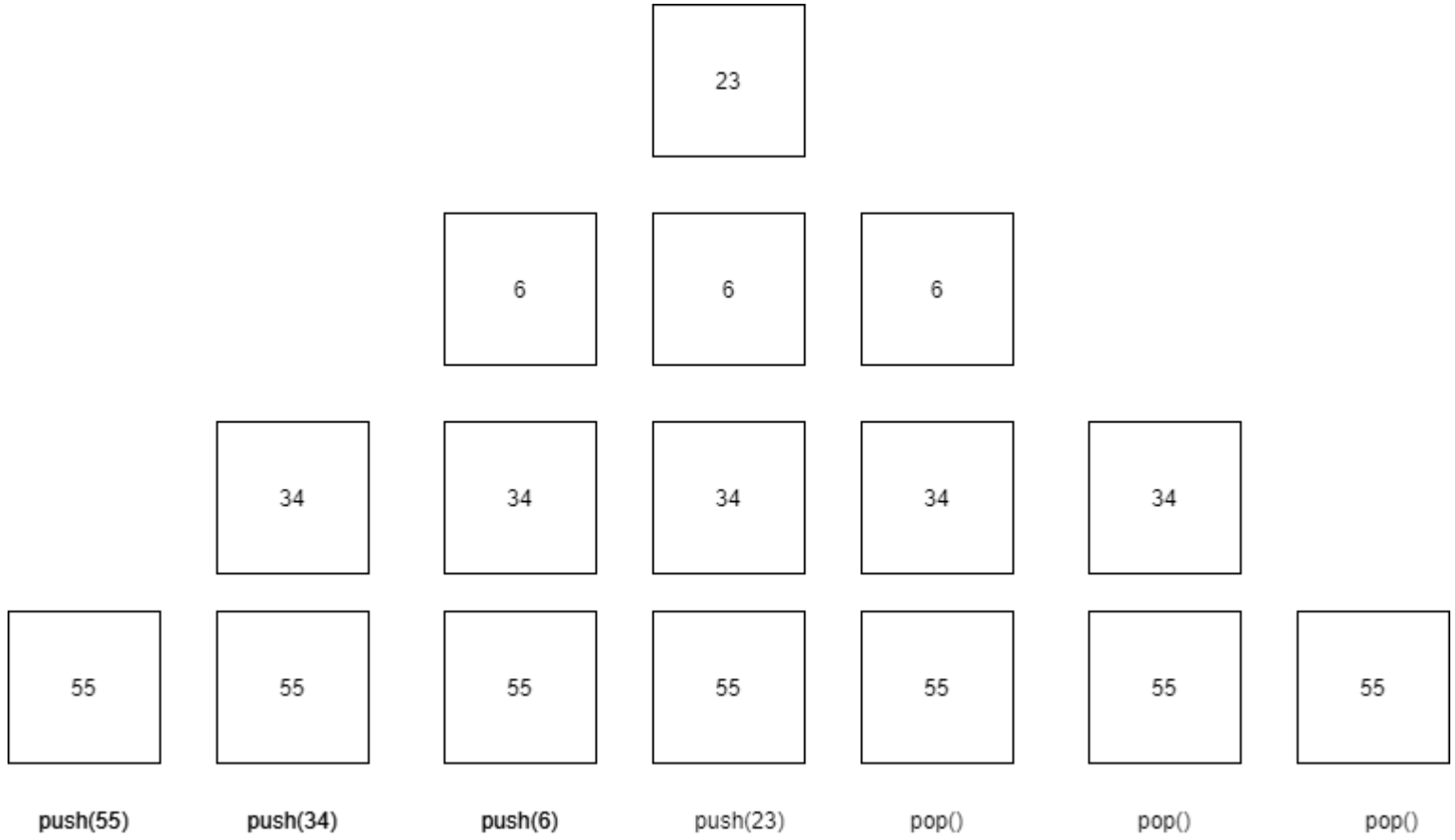
Yığın (Stack)



Abstract Data Type

- **Ana işlevler**
  - `void push()`
  - `T pop()`
- **Yardımcı işlevler**
  - `T top()`
  - `T size()`
  - `T isEmpty()`
  - `T isFull()`

Yığın (Stack)


23
6
6
6
34
34
34
34
34
55
55
55
55
55
55
55
push(55)
push(34)
push(6)
push(23)
pop()
pop()
pop()

Yığın (Stack)

- **Doğrudan kullanılan uygulamalar**
  - Sembollerin dengelenmesi (Balancing of symbols)
  - Fonksiyon çağrılarında (Function calls)
  - Dize ters çevirmede (String Reversal)
  - Infix-to-postfix dönüşümü
  - Postfix deyimlerinin değerlendirilmesi
  - Bir tarayıcıdaki ziyaret edilmiş sayfa listesi
  - Bir metin editöründe yapılan değişikliklerin geri alınması
  - HTML ve XML belgelerinde etiketlerin eşleşmesi
- **Dolaylı kullanılan uygulamalar**
  - Diğer veri yapılarında yardımcı veri yapısı olabilir, yani diğer veri yapılarının yardımcı bileşeni olarak kullanılabilir.
  - Tower of Hanoi, Tree Traversals, Stock span problem, histogram problem algoritmalarında kullanılabilir.
  - Backtracking algoritma tasarlama tekniğinde kullanılabilir.

Yığın (Stack)

- Basit dizi yapısı
- Dinamik dizi yapısı
- Bağlı liste uygulaması

Yığın (Stack)

- N elemana sahip olan bir yığın için:

Space complexity (for n push operations)	$O(n)$
Time complexity of createStack()	$O(1)$
Time complexity of push()	$O(1)$ (Average)
Time complexity of pop()	$O(1)$
Time complexity of top()	$O(1)$
Time complexity of isEmpty()	$O(1)$
Time complexity of deleteStack()	$O(n)$

Yığın (Stack) Kabakod

Pseudocode

```
Push(S,x)
    if Stack-Full(S)
    then error "overflow"
    else top(S) = top(S) + 1
         S[top(S)] = x
```

Yığın (Stack) Kabakod

Pseudocode

```
Pop(S)
    if Stack-Empty(S)
    then error "underflow"
    else top(S) = top(S) - 1
         return S[top(S) + 1]
```

Yığın (Stack) Kabakod

Pseudocode

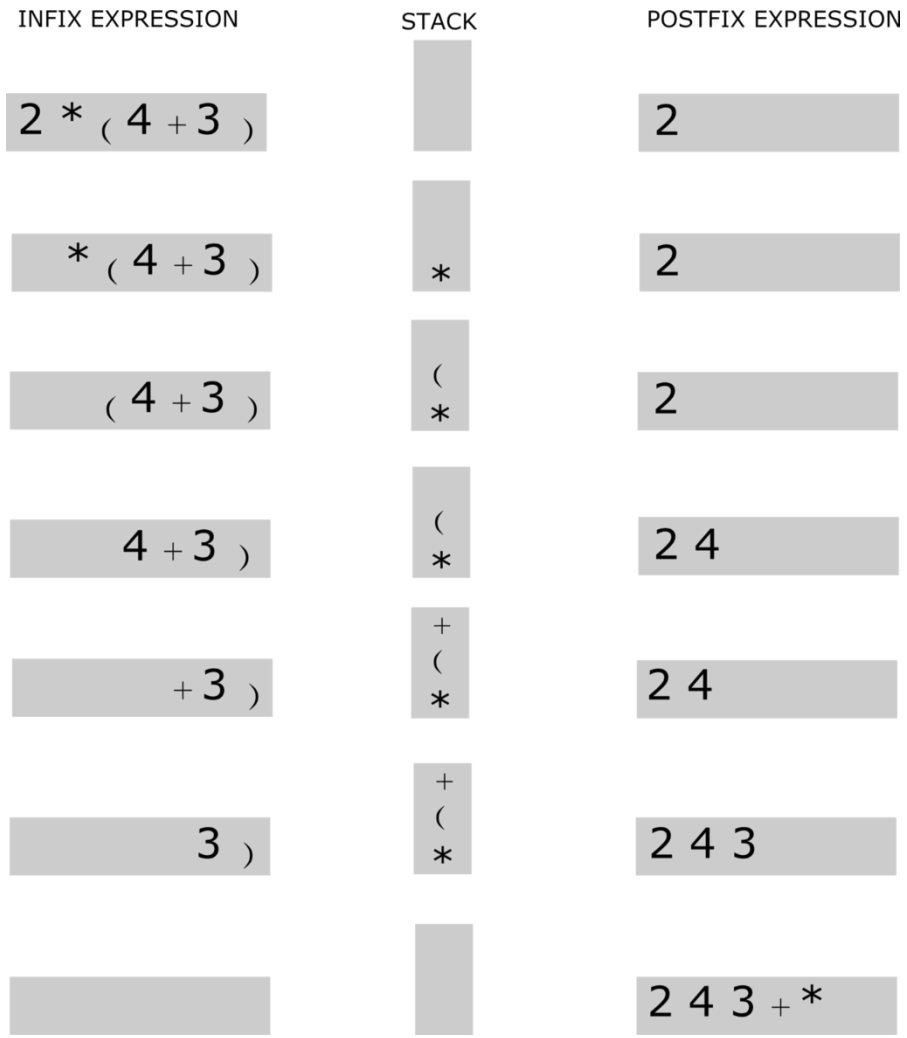
```
Stack-Empty(S)
    if top(S) = 0
    then return True
    else return False
```

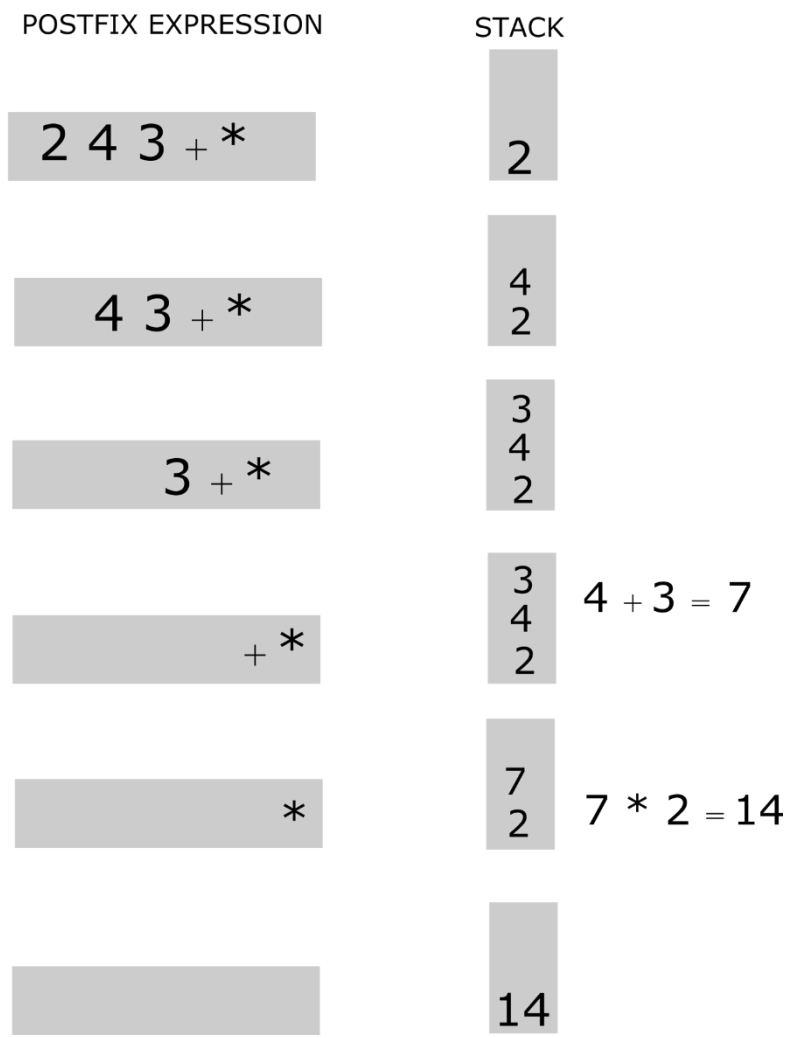
Yığın (Stack) Kabakod

Pseudocode

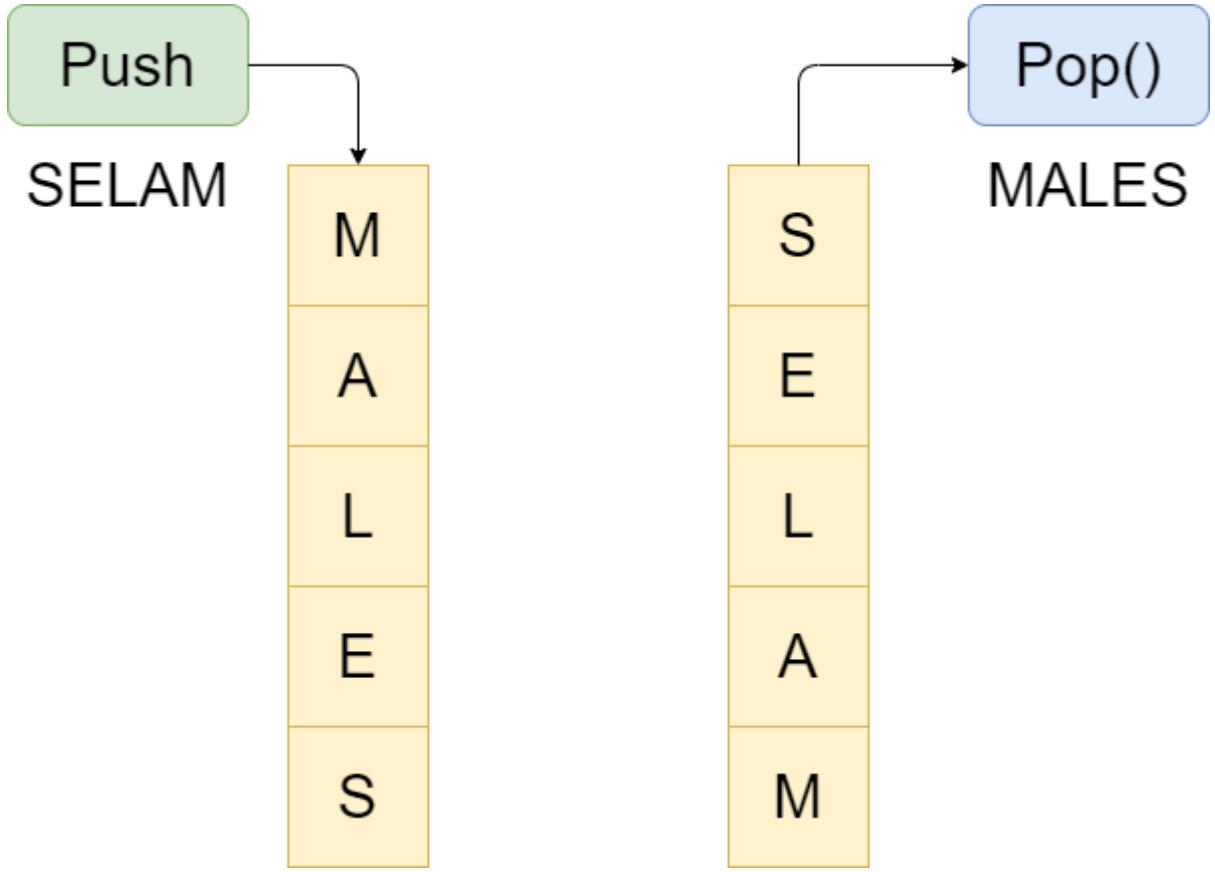
```
Stack-Full(S)
    if top(S) = length(S)
    then return True
    else return False
```

Infix to Postfix

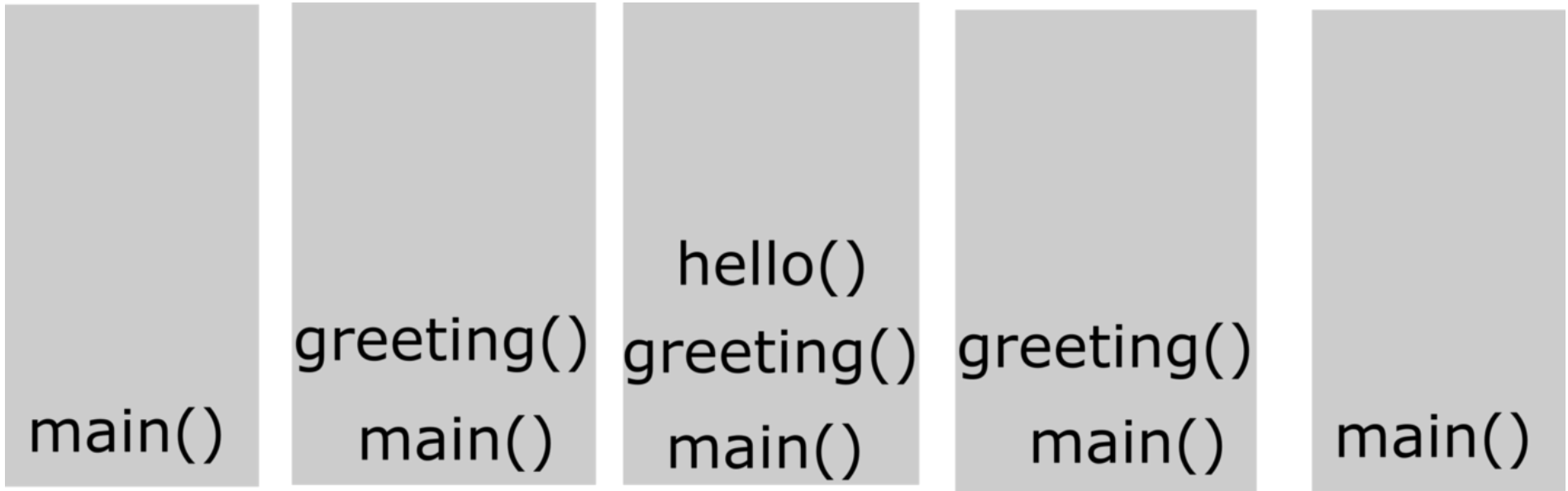


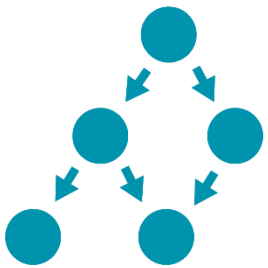


Reverse String



Function Calls





Veri Yapıları ve Algoritmalar

ZAFER CÖMERT
Öğretim Üyesi