

# Ozyineleme (Recursion) Nedir?

Calisma Kagidi

Ozyineleme, bir fonksiyonun bir problemi taban duruma ulasana kadar daha kucuk, ayni yapidaki alt problemlere bolmek icin kendini cagirmasidir; taban duruma ulasinca sonuclar geriye dogru birlestirilir.

## Sorular

- Ozyinelemeli bir fonksiyondaki taban durum nedir?
  - Yapilan ilk cagri
  - Daha fazla ozyinelemeli cagriyi durduran kosul
  - Fonksiyonun en yavas kismi
  - Bir sozdizimi hatasi
- Ozyinelemeli bir fonksiyonun taban durumu yoksa ne olur?
  - Daha hizli calisir
  - Sonsuz ozyinelemeye ve stack overflow'a yol acar
  - 10 cagri sonra otomatik durur
  - Hicbir sey olmaz, normal calisir
- $\text{factorial}(4) = 4 \text{ factorial}(3)$  ifadesinde  $\text{factorial}(3)$  neye denir?
  - Taban durum
  - Ozyinelemeli cagri
  - Return ifadesi
  - Sonsuz dongu
- $F(n) = F(n-1) + F(n-2)$ ,  $F(0)=0$ ,  $F(1)=1$  ise  $F(4)$  kac tir?
  - 2
  - 3
  - 5
  - 8
- $\text{factorial}(5)$ 'i ozyinelemeli olarak hesapla.
- $[3, 5, 2]$  listesinin toplamini ozyinelemeyle bul.
- $F(n) = F(n-1) + F(n-2)$ ,  $F(0)=0$ ,  $F(1)=1$  tanimiyla Fibonacci(5)'i bul.
- Tanimla: Ozyineleme (recursion) nedir?
- Tanimla: Ozyinelemeli bir fonksiyonun iki temel parcasi nedir?
- Tanimla: Taban durum olmazsa ne olur?

## Cevap Anahtari

1. B) Daha fazla ozyinelemeli cagriyi durduran kosul - Taban durum, baska bir ozyinelemeli cagri olmadan dogrudan cevaplanabilen en basit girdidir ve zinciri durdurur.
2. B) Sonsuz ozyinelemeye ve stack overflow'a yol acar - Taban durum olmadan fonksiyon kendini cagirmayi asla birakmaz ve cagri yiginini tuketir.
3. B) Ozyinelemeli cagri - factorial(3), fonksiyonun kendisini daha kucuk bir girdiyle cagirmasi olan ozyinelemeli cagridir.
4. B)  $3 - F(2)=1, F(3)=2, F(4)=F(3)+F(2)=2+1=3$ .
5. factorial(5) = 5 factorial(4) factorial(4) = 4 factorial(3) factorial(3) = 3 factorial(2) factorial(2) = 2 factorial(1) factorial(1) = 1 factorial(0) = 1 1 = 1 Geri sarma:  $2! = 2, 3! = 6, 4! = 24, 5! = 120$  factorial(5) = 120
6.  $sum([3,5,2]) = 3 + sum([5,2])$   $sum([5,2]) = 5 + sum([2])$   $sum([2]) = 2 + sum([]) = 2 + 0 = 2$  Geri sarma:  $5+2=7, 3+7=10$  Sonuc: 10
7.  $F(2) = F(1)+F(0) = 1+0 = 1$   $F(3) = F(2)+F(1) = 1+1 = 2$   $F(4) = F(3)+F(2) = 2+1 = 3$   $F(5) = F(4)+F(3) = 3+2 = 5$  Sonuc:  $F(5) = 5$
8. Bir fonksiyonun ayni problemin daha kucuk orneklerini cozme icin kendini cagirma teknigidir.
9. Taban durum (ozyinelemeyi durdurur) ve ozyinelemeli durum (kendini daha kucuk bir girdiyle cagirir).
10. Sonsuz ozyineleme olusur ve sonunda 'stack overflow' (yigin tasmasi) hatasiyla cokur.

### Bounlu

Tum kartlar, adim adim cozumler ve AI hoca destegi Notek uygulamasinda.  
Sinav tarihlerini Promy otomatik hatirlaticiya ceviris.