

# Greedy (Açgözlü) Yaklaşım

→ Optimizasyon problemleri için bir çözüm yöntemidir.

→ Dinamik Programlama zıttı bir şekilde yukarıdan aşağıya çalışır. (Top-Down)

→ Her zaman en iyi çözümü bulamaz.

Bottom up çalışır ve en iyi çözümü gösterir.

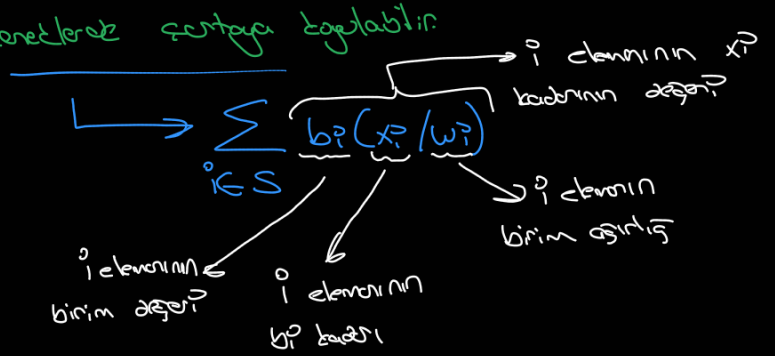
## Kesirli (Fractional Knapsack) Sırt Çantası Problemi.

→ Burada materyalın yarı, çeyresi gibi kesirlerle sığdırılabilir.

→ Brute Force ile maliyet ( $2^n$ ) olur.

→ Dinamik Programlama ile ( $n \cdot w$ ) olur.

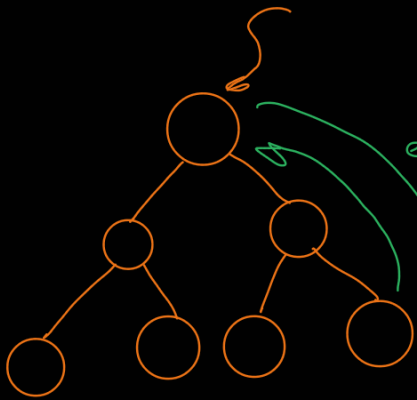
→ Bu yaklaşımla ( $n \log n$ ) sınıfı olur.



\* Çantanın güncel ağırlığı maks ağırlıktan az olduğu sürece, en değerli eşyayı

Seç ve çantaya sığacak kadar kısmını çantaya ekle → Greedy Choice

→ Bu işlem için PR (priority Queue) (öncelikli kuyruk) kullanılır.



en üstteki ile değişim maliyeti ( $\log n$ )

→ bu işlem n eleman için

çalıştırıldığında maliyet maksimum  $(n \log n)$  olacaktır.  
Zaman Maliyeti:

→  $O(n \log n)$

→ kesirli Sırt Çantası problemi, as güzel seçim özelliği ile çözüm.

\* 0-1 Sırt Çantası için Greedy yaklaşım uygun bir çözüm değildir.

→ olgun nalıyet dir.

→ Top çok yaklaşım kullanır ve her zaman en iyi çözümü vermez.

## Greedy Yaklaşım ile Çözüm

→  $O$  en çok a değeri element alınarak işlem yapılır

→ Big  $O$  notasyona göre  $O(n \log n)$  zaman nalıyetinde dir.

Bu değerler güçlü gibi gözükmektedir

Anlatım = Eşyaların ağırlıkları ve değerlerinin doğru kümesini, değerlerine göre büyükten küçüğe dizliyoruz. Burada bir sıralama algoritması salımalı ve nalıyetimize sıralama algoritmasının nalıyeti de edenecek sıralama yapılarına sıra ile element çantanın kapasitesine uygunca çanta yerleştirilir ve çanta kapasitesi uygun olmayınca keser en çok değer ile element çanta yerleştirilir. Çanta kapasitesi doğru keser işlem devam ettirir.

Örnek  $I = I_1, I_2, I_3, I_4, I_5 \rightarrow$  Eşyalar  
 $w = 5, 10, 20, 30, 40 \rightarrow$  Ağırlıklar  
 $v = 30, 20, 100, 90, 160 \rightarrow$  Değerleri

$W = 60 \rightarrow$  Çanta kapasitesi

(-) Yaklaşımın tespit edilir.

$I_1 = \frac{30}{5} = 6$   
 $I_2 = \frac{20}{10} = 2$   
 $I_3 = \frac{100}{20} = 5$   
 $I_4 = \frac{90}{30} = 3$   
 $I_5 = \frac{160}{40} = 4$

Bu değerler kümesi

$U' = (6, 2, 5, 3, 4) \rightarrow$  Bu küme büyükten küçüğe dizilir

$U' = (6, 5, 4, 3, 2) \rightarrow$  Büyük değerden çanta kaynağı başlanır çanta ağırlık olarak doğru keser devam

4 değerli element çanta kaynağı başlanır çanta tespit kesirli doğru seçim keser keser 4 değerli element kaynağı başlanır doğru keser

Çanta  $4 = I_5 \rightarrow w = 40 \rightarrow v = 160$   
 kalan kapasite  $= 35 \text{ kg}$   
 $160/35$

$$\rightarrow 30 \times \frac{5}{5} + 100 \times \frac{20}{20} + 160 \times \frac{37}{40} = \text{Value} = 270$$

$$30 + 100 + 140 = 270$$

\* A<sub>5</sub> gözüyle yatkın optimal çözümü garanti etmez

☉ Para izi problemi

① Ödenmesi gereken miktar 24 ₺'dir. Para değerlerimiz 20, 10, 5, 1 olsun

→ 24'ü ele almak için algoritma önce 20'yi seçer, kalan 4'ü tamamlamak

için 4 defa 1'ı seçer ve 5 adanmış olur. Maliyet 5.

Ama bir 10 ve bir 5 ile 2 maliyet ile çözüldü