CENG 213 Veri Yapıları 7: Binary Heap ve Huffman Ağaçları

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

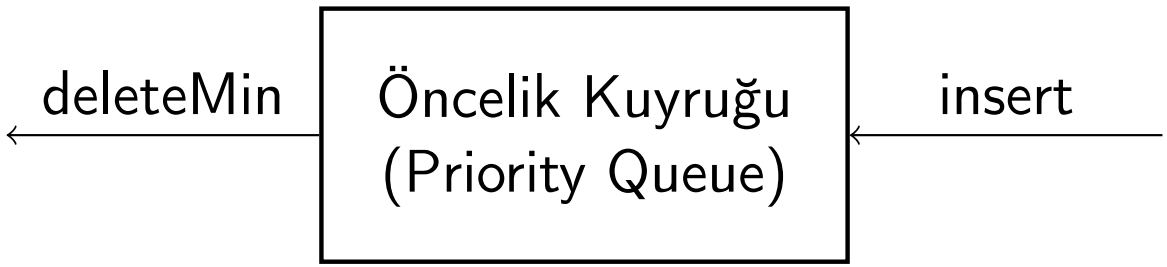
Hafta 7

Anahat

Öncelik Kuyruğu(Priority Queue) Tanım

Tanım

Öncelik kuyruğu ekleme(insert) ve en küçüğü silme(deleteMin) işlemlerine sahip bir veri yapısıdır. deleteMin işlemi normal kuyruk yapısındaki dequeue işlemi gibi çalışır ama her zaman en küçük değere sahip elemanı listeden çıkarır.



Şekil: Öncelik kuyruğu modeli

İkili Yığın/Binary Heap

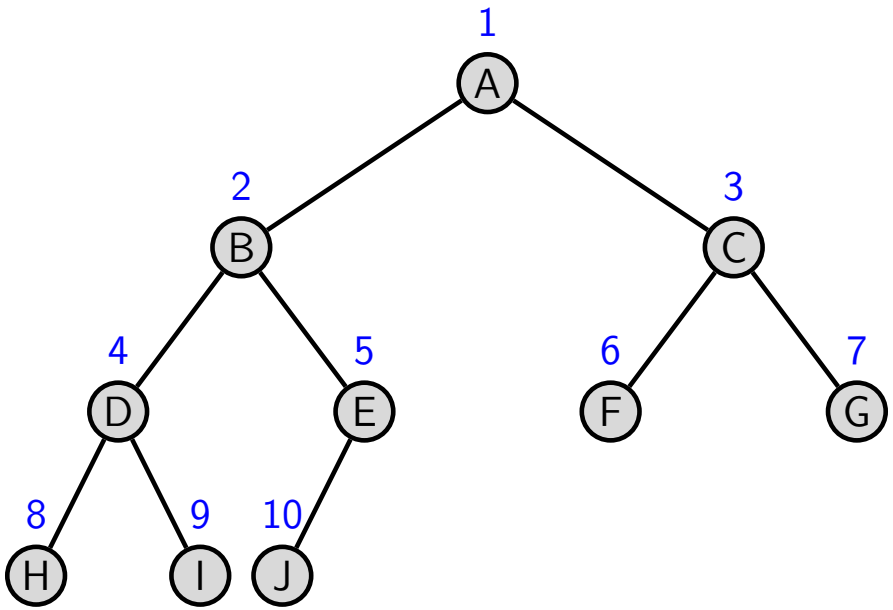
Tanım

Öncelik kuyruklarının gerçekleştirilmesi için ikili ağaç yapısında verimli bir veri yapısıdır.

Heap aşağıdaki özellikleri sağlar:

- Heap tam ikili ağaç(complete binary tree) yapısındadır
- Bir düğümdeki anahtar değeri çocukarının anahtar değerinden daha küçüktür(Minimum Heap)
- Tam ikili ağaç yapısında olduğu için dizilerle temsil edilebilir, sol çocuk: $2i$, sağ çocuk: $2i + 1$
- Ekleme ve silme işlemlerinde bozulan heap yapısını oluşturmak için **Heapify** işlemi yapılır
- Minimum ya da maksimum türleri olabilir.
  - Maksimum heap yapısında kökte en büyük değer yer alır ve her düğümün çocukları kendisinden küçüktür.

Heap Örneği



Şekil: Heap Örneği

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Değer	Boş	A	B	C	D	E	F	G	H	I	J			

Ekleme

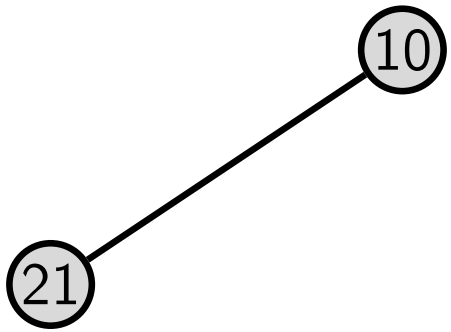
- Ekleme işlemi tam ikili ağacın en alt seviyesinin soldaki ilk boşluğuna yerleştirilerek yapılır
- Eğer Heap özelliği bozulmuşsa ebeveyn ile eklenen çocuk yer değiştirir, bu işlem köke kadar devam eder

Ekleme Örneği: 10

~~10~~, 21, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim

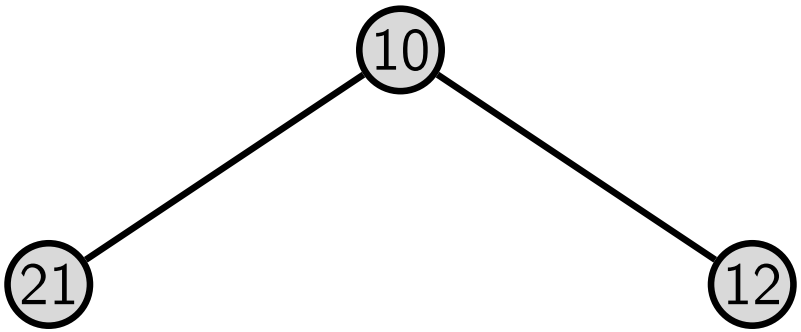
Ekleme Örneği: 21

~~10~~, ~~21~~, 12, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



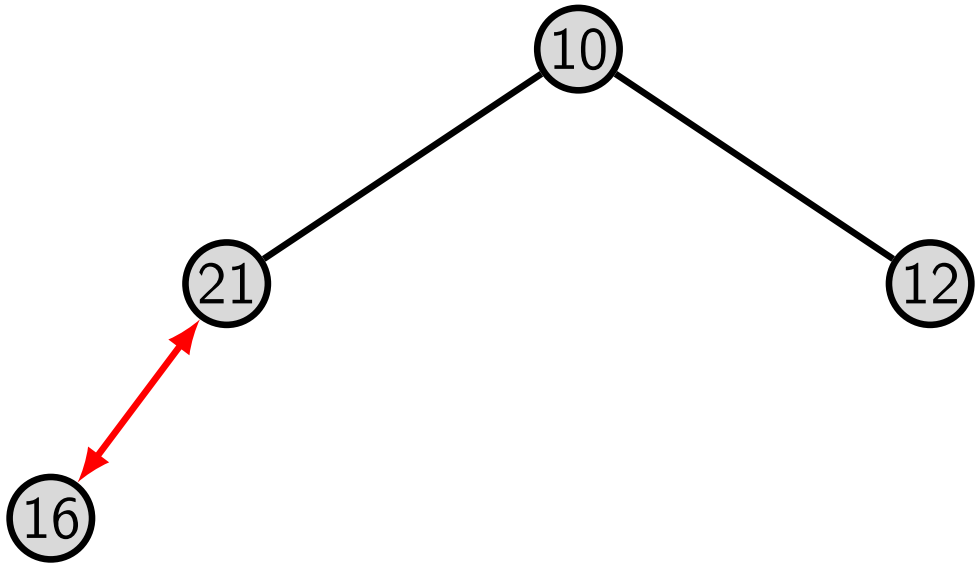
Ekleme Örneği: 21

~~10~~, ~~21~~, ~~12~~, 16, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



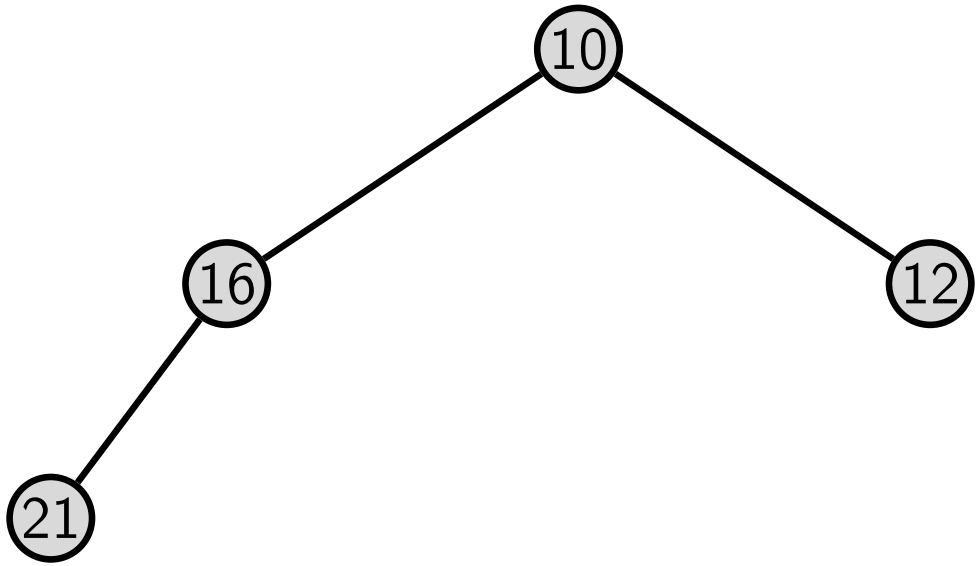
Ekleme Örneği: 16

~~10~~, ~~21~~, ~~12~~, ~~16~~, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



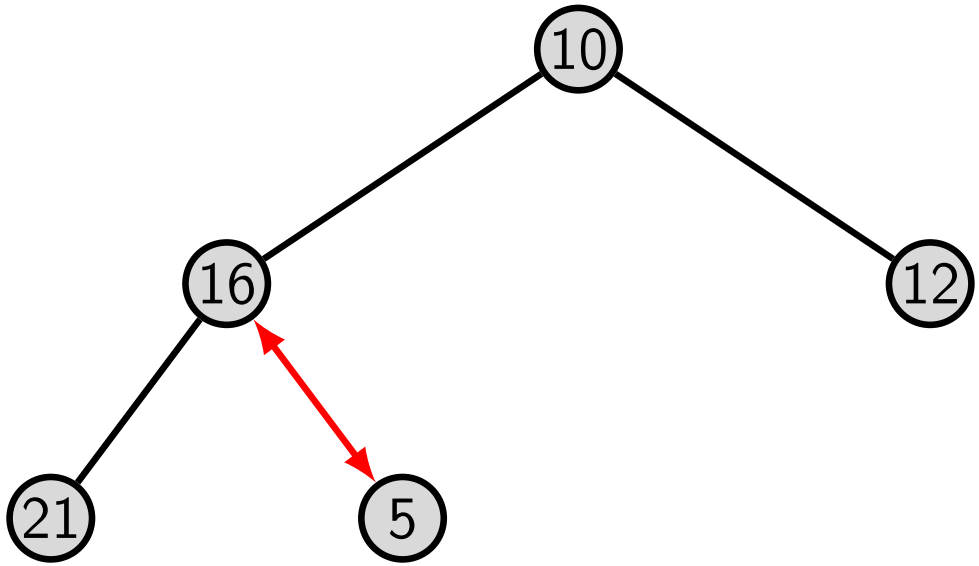
Ekleme Örneği: 16

~~10~~, ~~21~~, ~~12~~, ~~16~~, 5, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



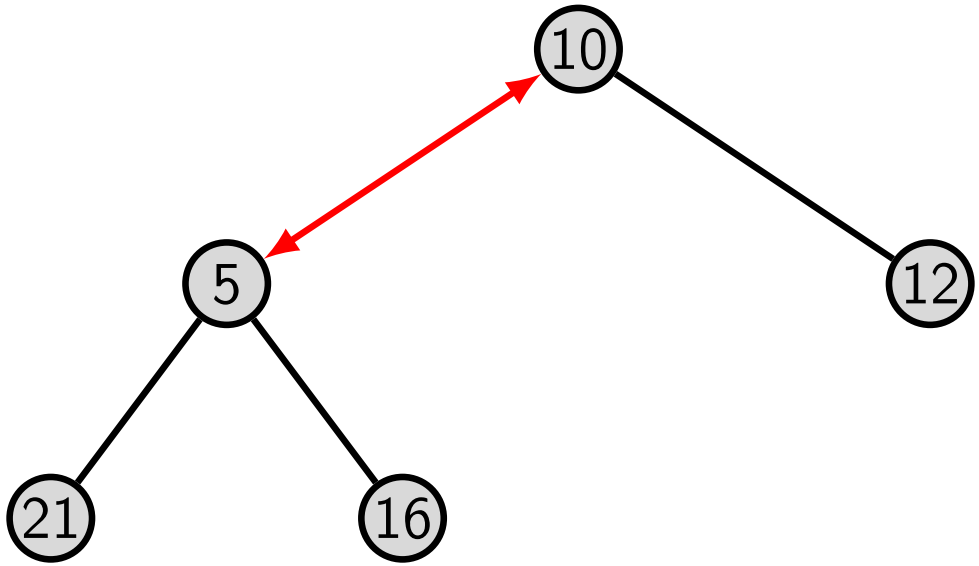
Ekleme Örneği: 5

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



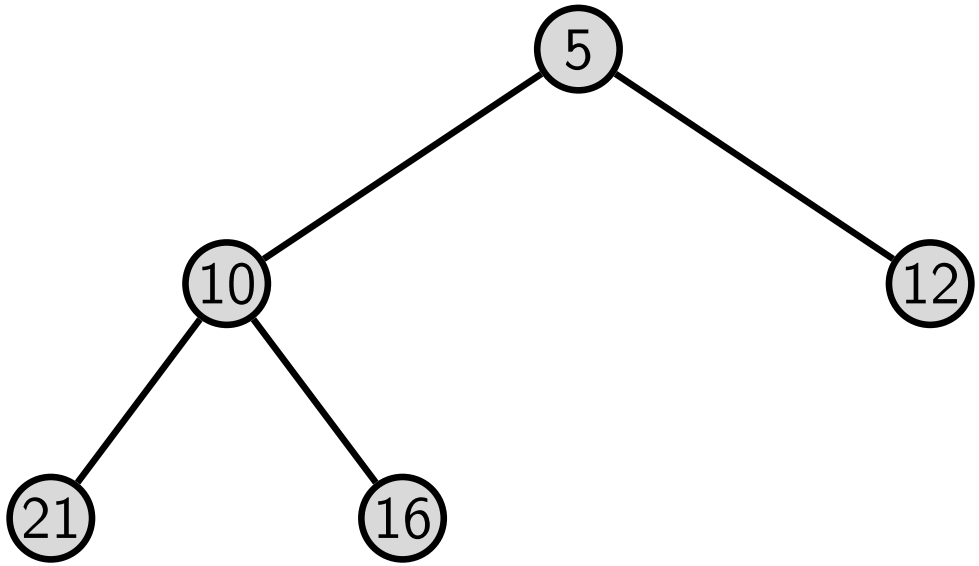
Ekleme Örneği: 5

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



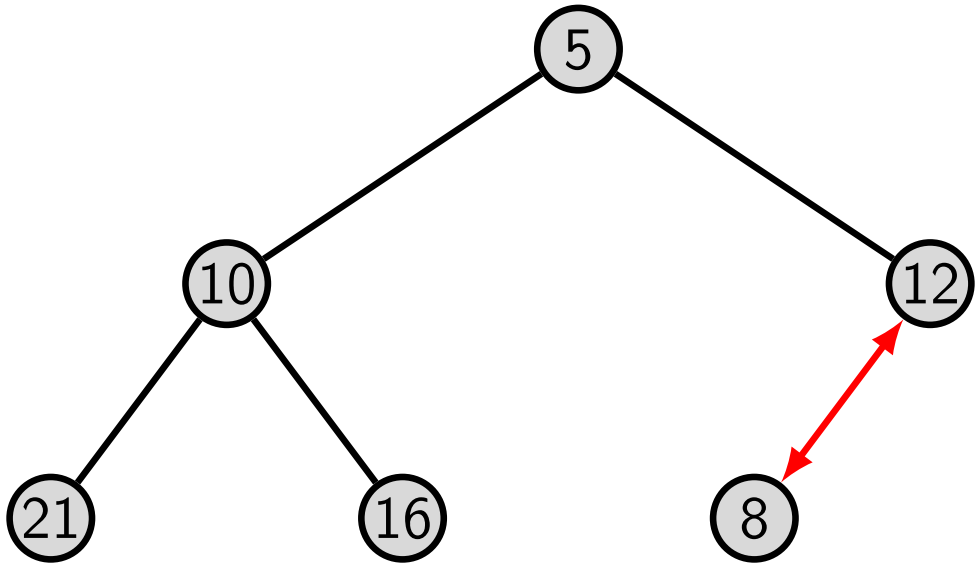
Ekleme Örneği: 5

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, 8, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



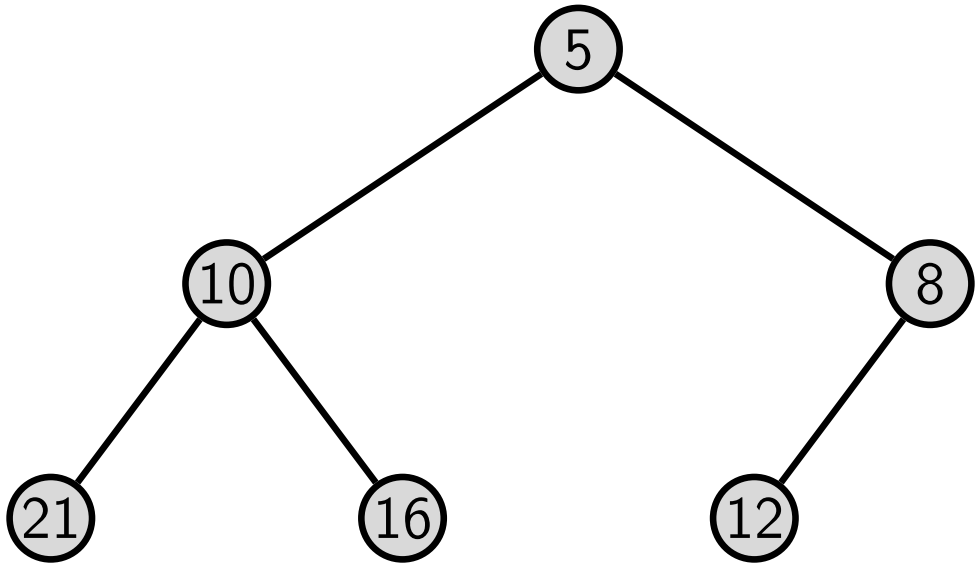
Ekleme Örneği: 8

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



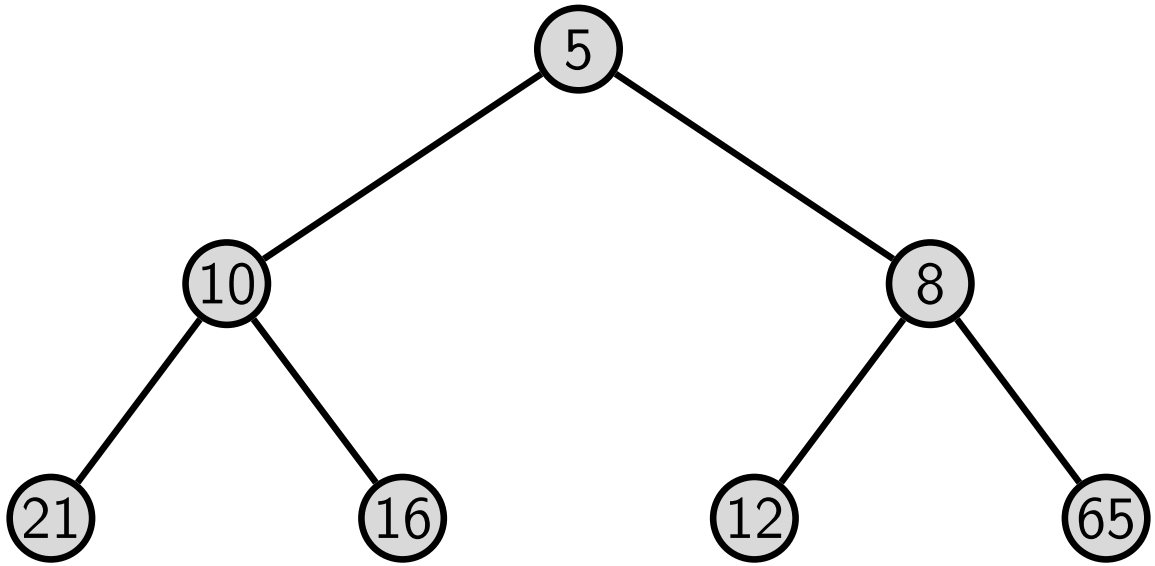
Ekleme Örneği: 8

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, 65, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



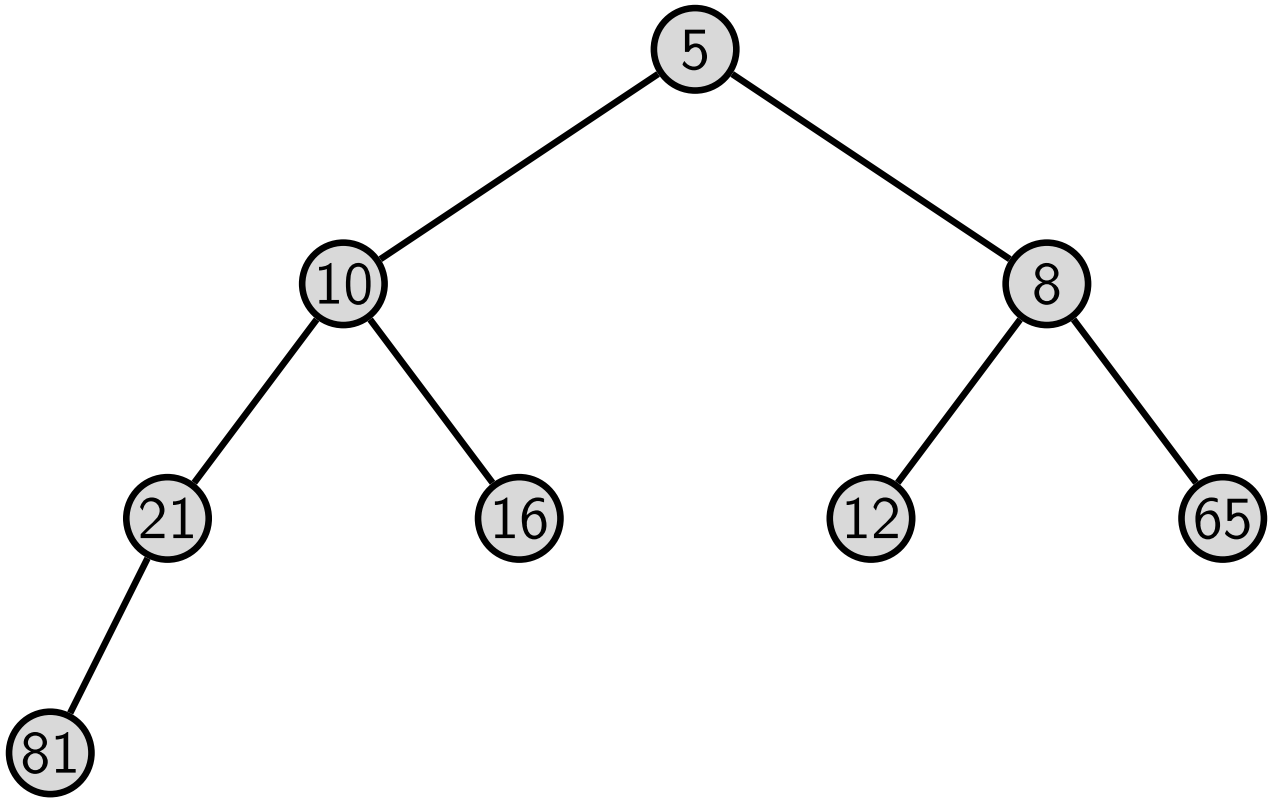
Ekleme Örneği: 65

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, 81, 15, 4 değerlerini boş bir ikili yığına ekleyelim



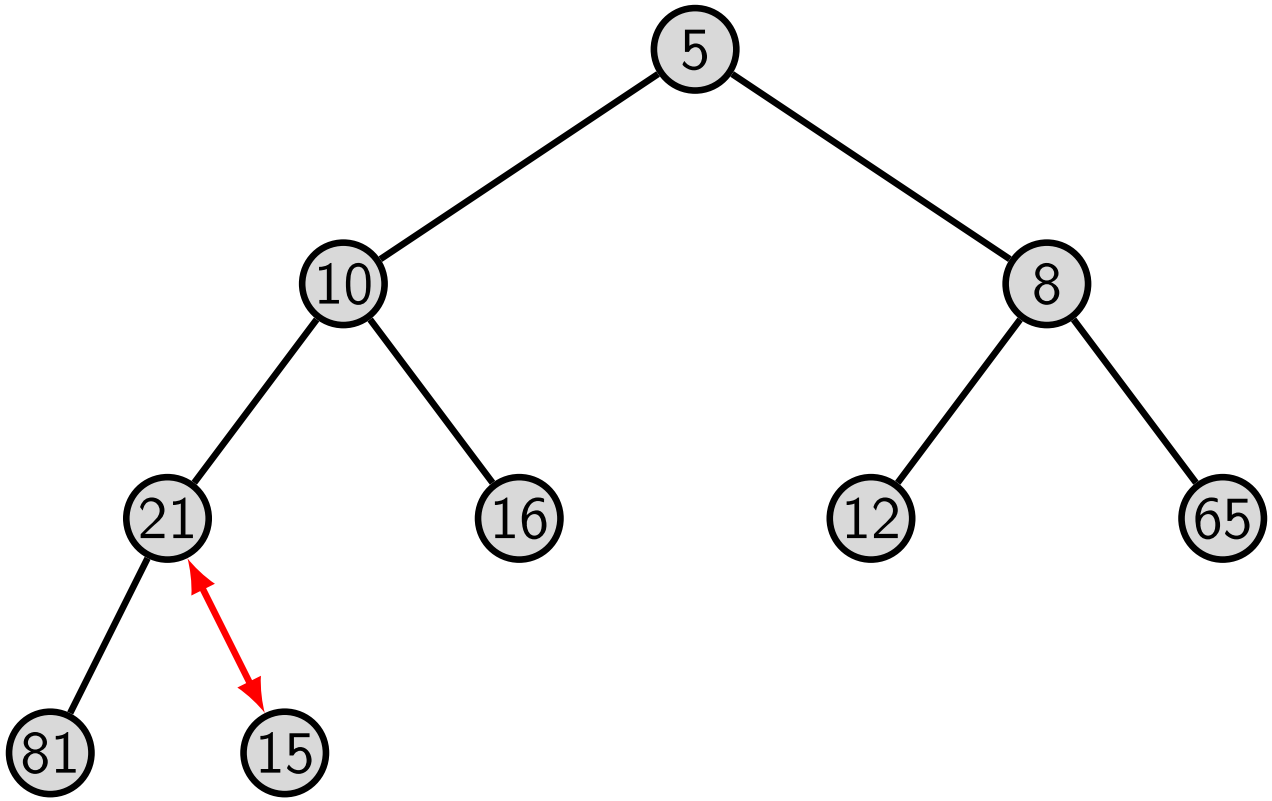
Ekleme Örneği: 81

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, 15, 4 değerlerini boş bir ikili yığına ekleyelim



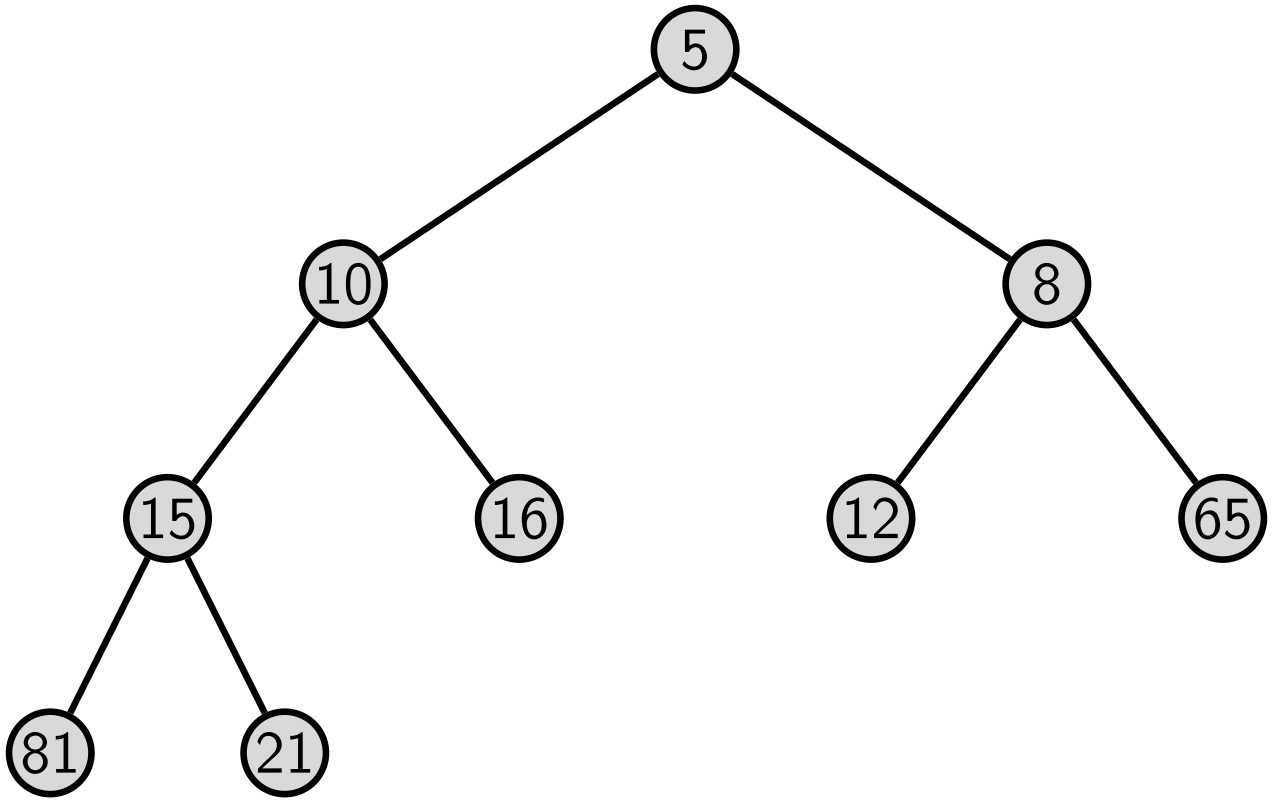
Ekleme Örneği: 15

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, 4 değerlerini boş bir ikili yığına ekleyelim



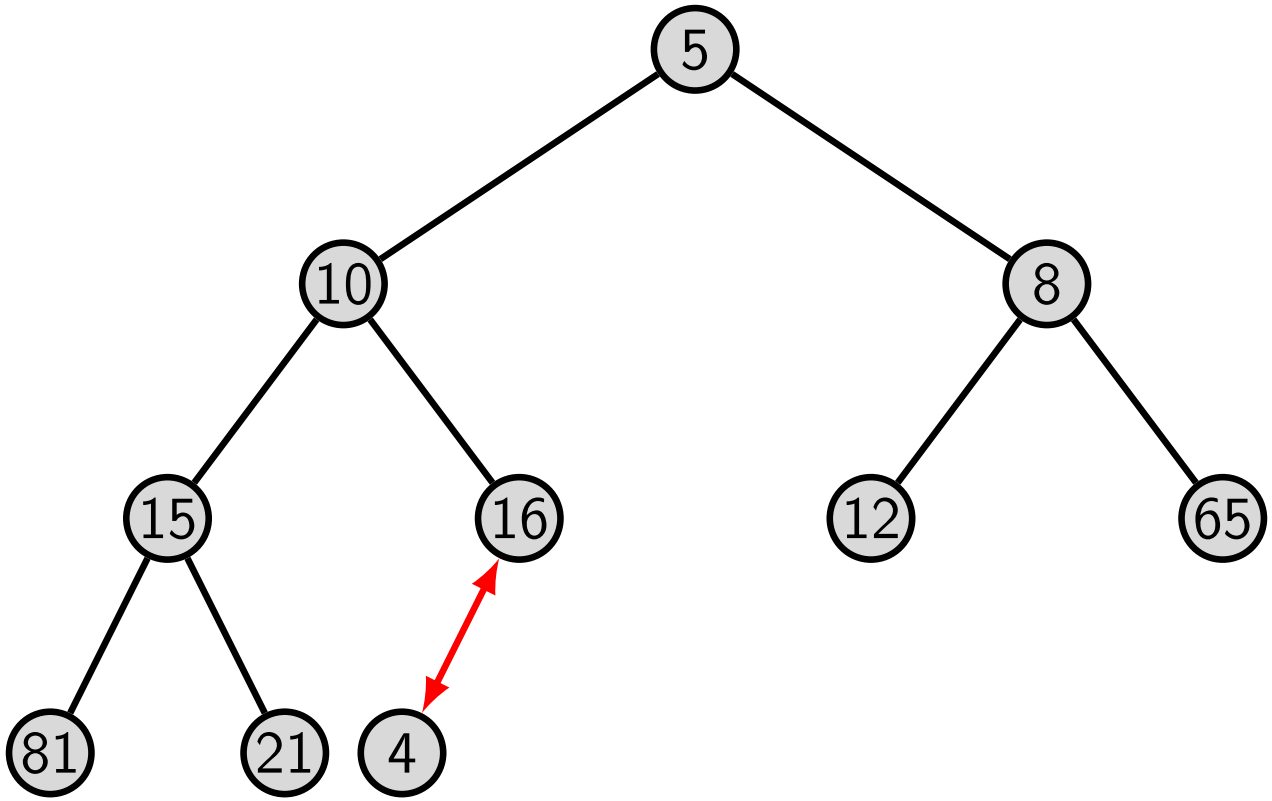
Ekleme Örneği: 15

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, 4 değerlerini boş bir ikili yığına ekleyelim



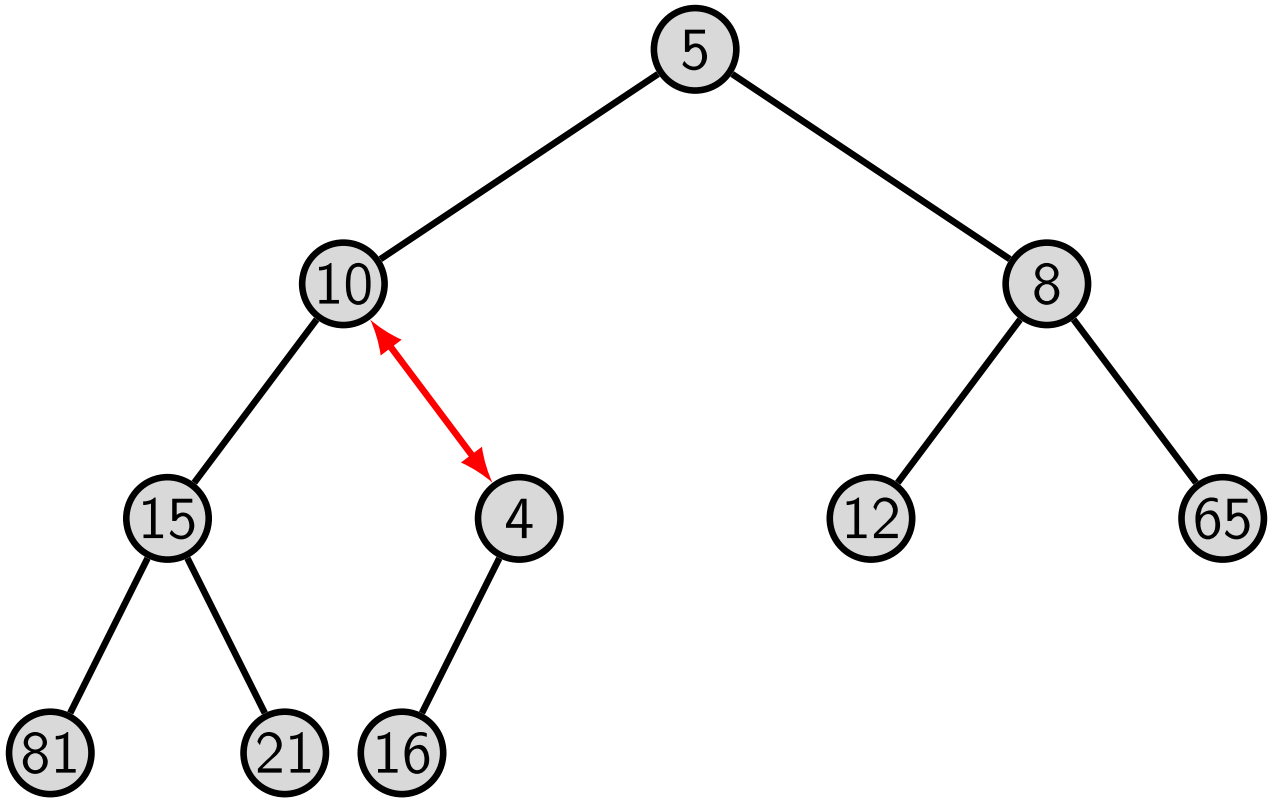
Ekleme Örneği: 4

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, ~~4~~ değerlerini boş bir ikili yığına ekleyelim



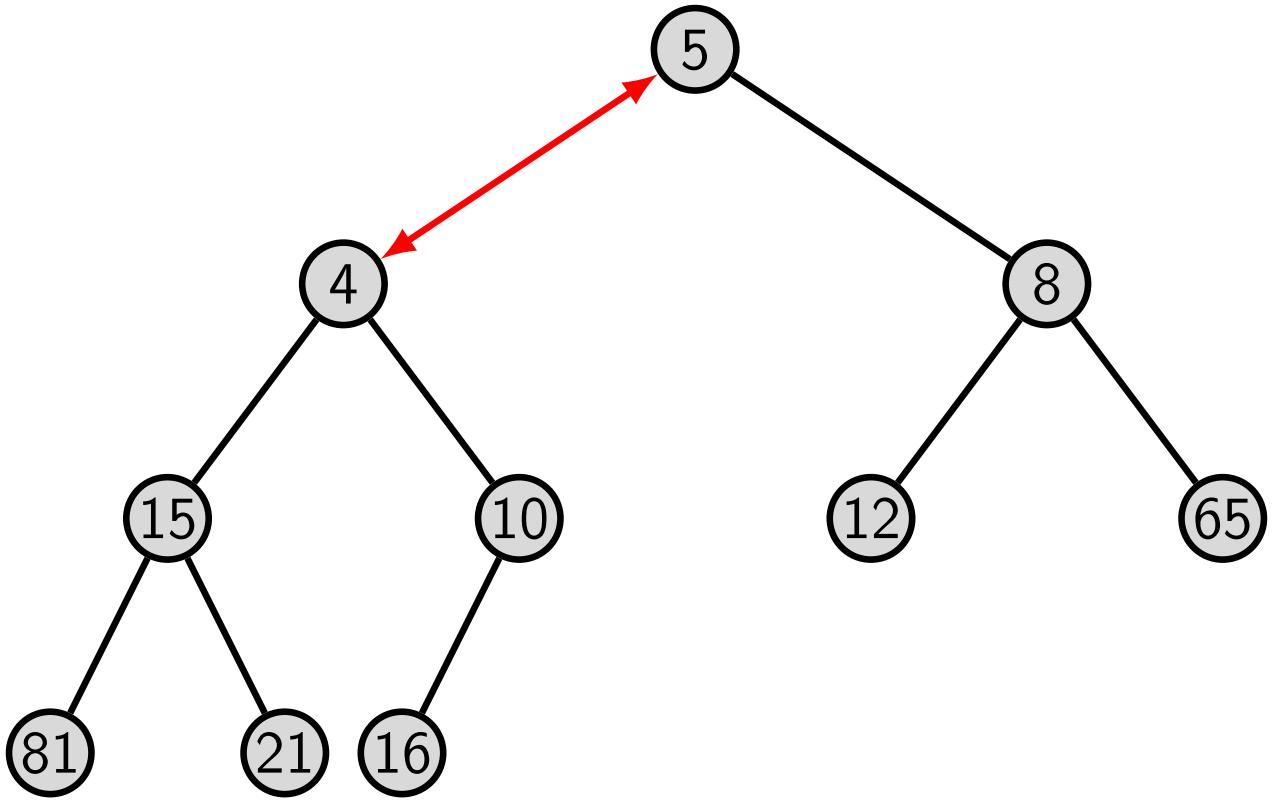
Ekleme Örneği: 4

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, ~~4~~ değerlerini boş bir ikili yığına ekleyelim



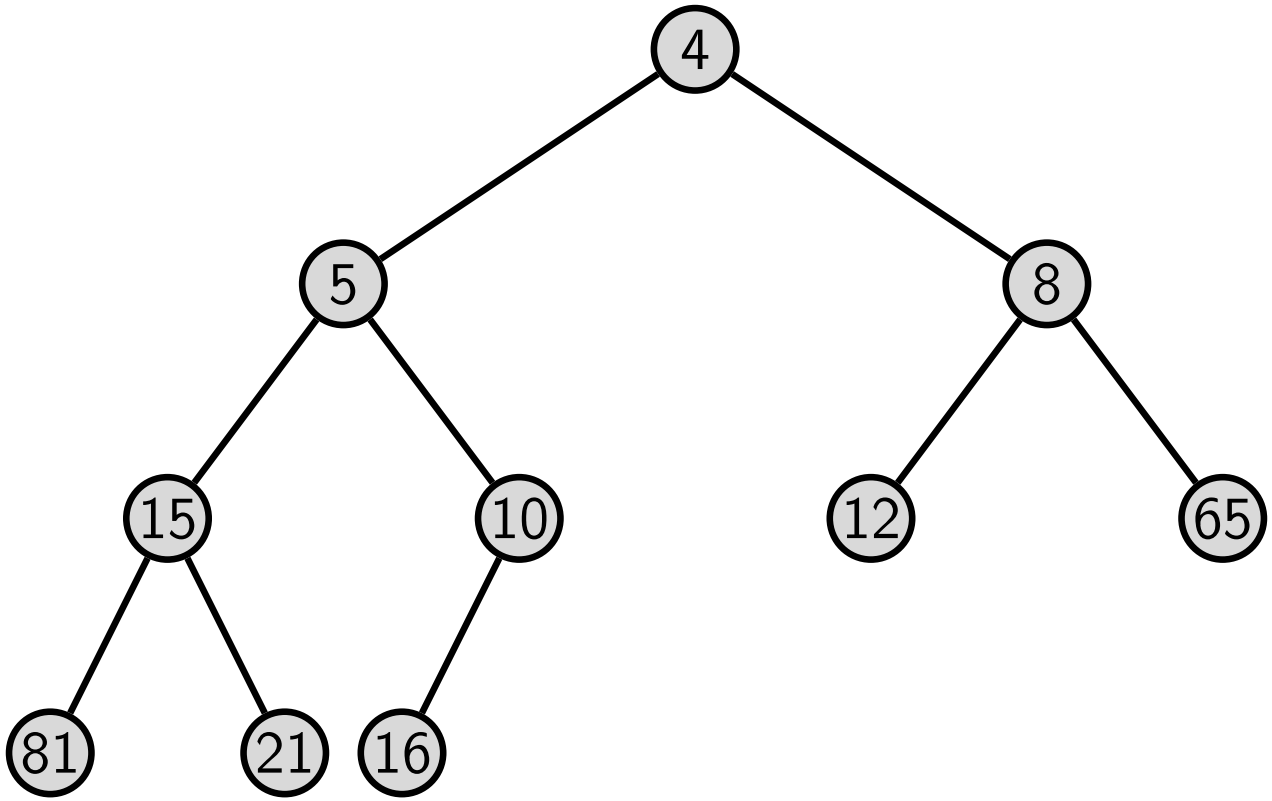
Ekleme Örneği: 4

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, ~~4~~ değerlerini boş bir ikili yığına ekleyelim



Ekleme Örneği: 4

~~10~~, ~~21~~, ~~12~~, ~~16~~, ~~5~~, ~~8~~, ~~65~~, ~~81~~, ~~15~~, ~~4~~ değerlerini boş bir ikili yığına ekleyelim

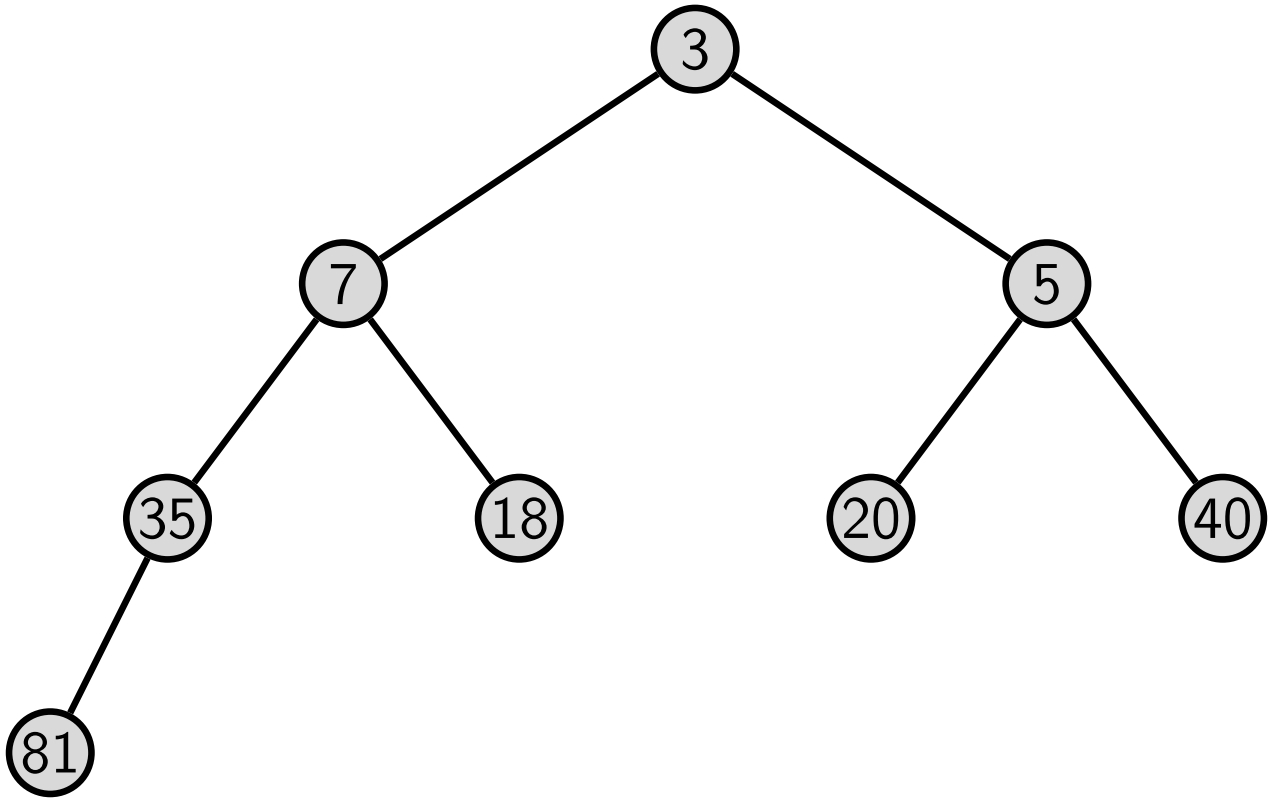


Soru

5, 18, 20, 35, 7, 3, 40, 81 değerlerini sırasılya boş bir ikili yığına ekleyin ve ağacı çizin.

Soru

5, 18, 20, 35, 7, 3, 40, 81 değerlerini sırasılya boş bir ikili yığına ekleyin ve ağacı çizin.

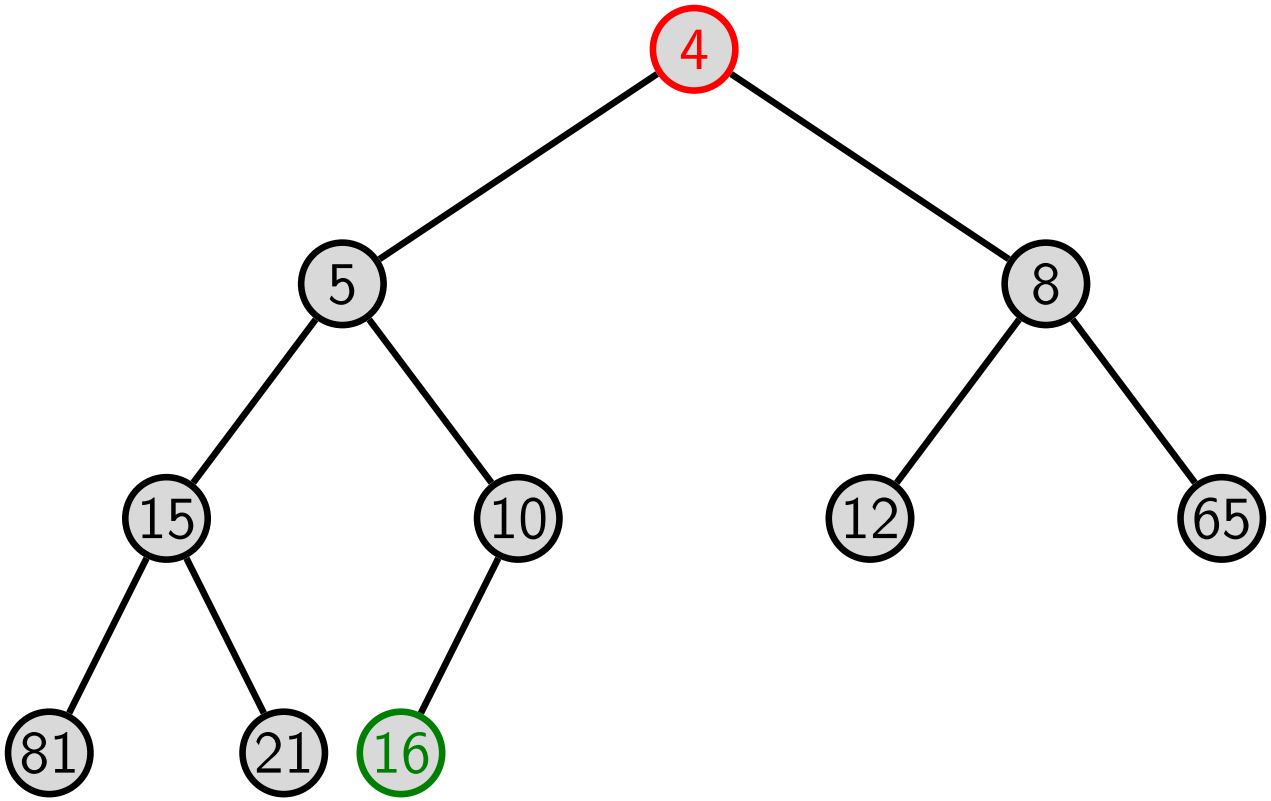


Silme

- Silme işlemi kökten yapılır(deleteMin)
- Kökteki elemanın yerine en alt seviyenin en sağındaki eleman çıkartılıp yazılır
- Heap özelliğini sağlamak için aşağı doğru çocuklarından hangisi küçükse onunla yer değiştirir

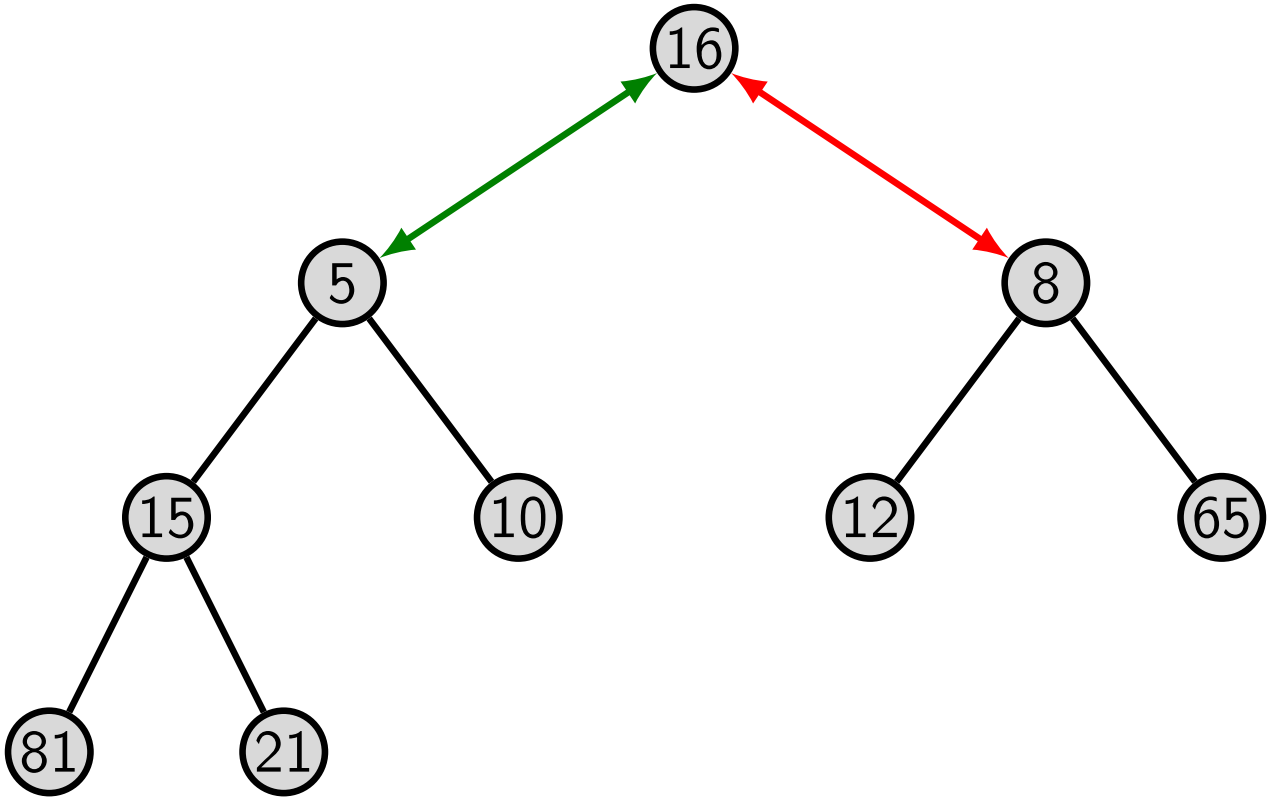
Silme Örneği

Kökteki 4 değerini silme



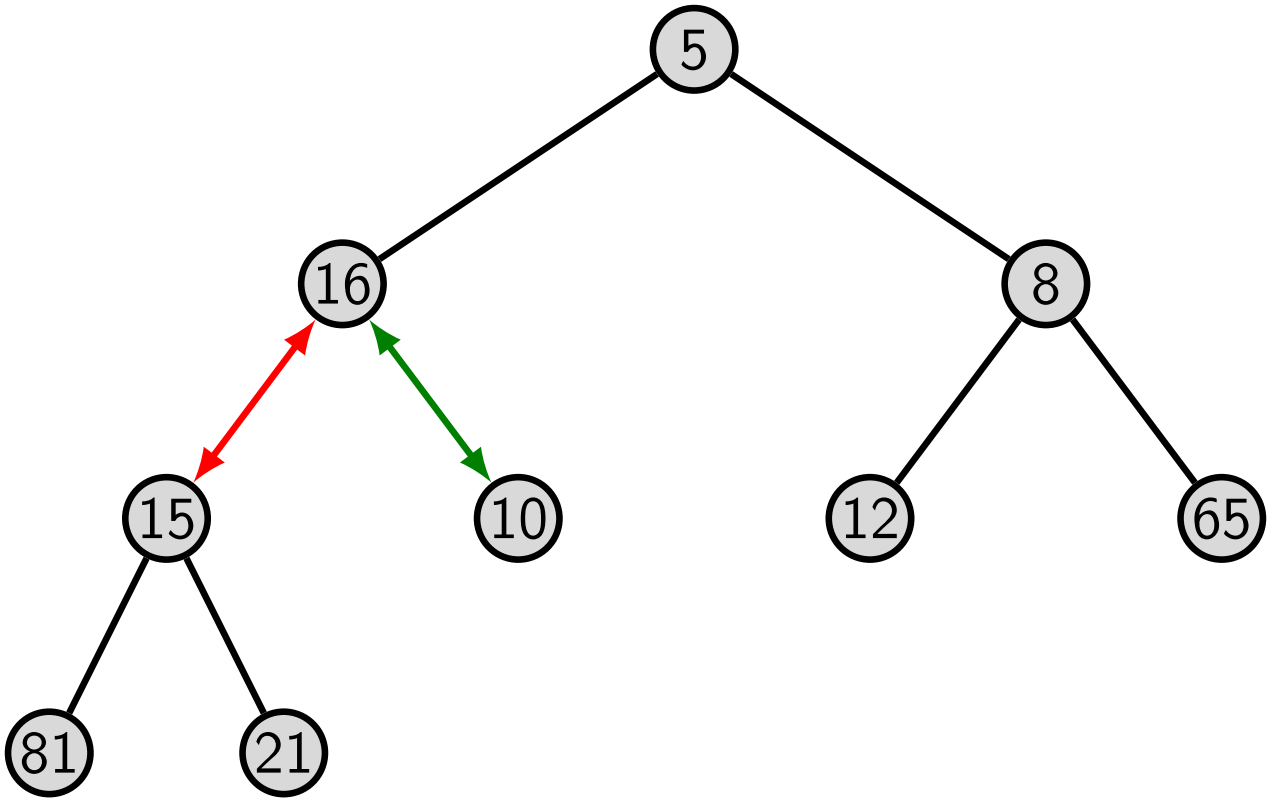
Silme Örneği

Kökteki 4 değerini silme



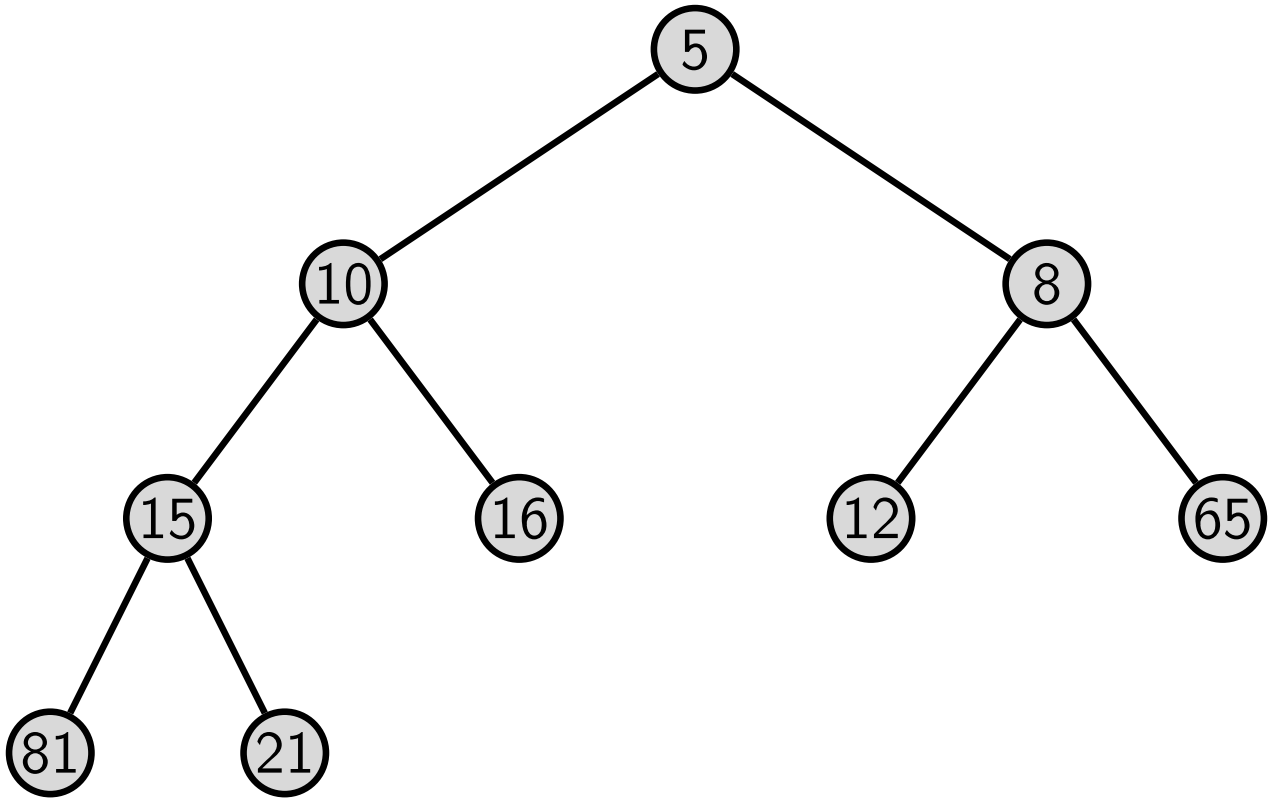
Silme Örneği

Kökteki 4 değerini silme



Silme Örneği

Kökteki 4 değerini silme



Ağaç İşlemleri Zaman Karmaşıklıkları(En kötü durum)

Tablo: Temel işlemlerin zaman karmaşıklıkları

Data Structure	**Search**	**Insert**	**Delete**
BST	O(n)	O(n)	O(n)
AVL	O(log n)	O(log n)	O(log n)
B-Tree	O(log n)	O(log n)	O(log n)
Heap	O(n)	O(log n)	O(log n)

Görselleştirme

- İkili Yığın(Binary Heap) için
  http://www.cs.usfca.edu/ galles/visualization/Heap.html

Huffman Kodlama

Huffman kodlama, kayıpsız bir veri sıkıştırma yöntemidir. Genellikle bir metin içindeki karakterler ikili(binary) bir koda dönüştürülür ve bu kodun boyutunun orjinal metnin boyutundan daha küçük olması beklenir. Tüm karakterlerin frekans(belirme sayısı) değeri hesaplanır ve bu değerler kullanılarak ikili bir ağaç oluşturulur.

Huffman Ağacı Oluşturma Algoritması

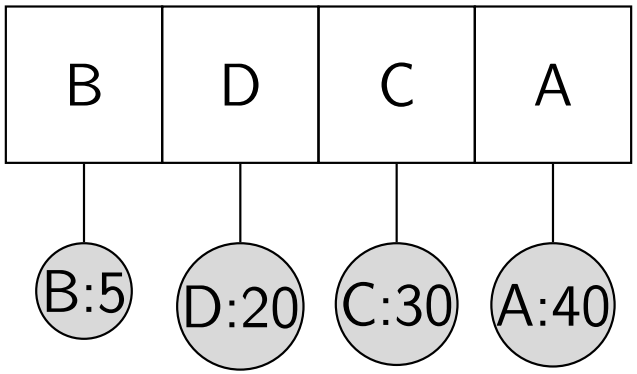
Düğümleri saklamak için öncelik kuyruğu oluştur(pq)
Bütün karakterleri frekans değerine göre kuyruğa ekle
while *Kuyrukta birden fazla eleman olduğu sürece* **do**
    n1=Kuyruktan bir düğüm çek
    n2=Kuyruktan bir düğüm çek
    Sol çocuğu n1, sağ çocuğu n2 ve frekansı toplamları olacak şekilde yeni bir düğüm oluştur ve kuyruğa ekle
end
Kuyruktaki tek değer ağacın kökünü verir

Algorithm 1: Huffman Ağacı Oluşturma Algoritması

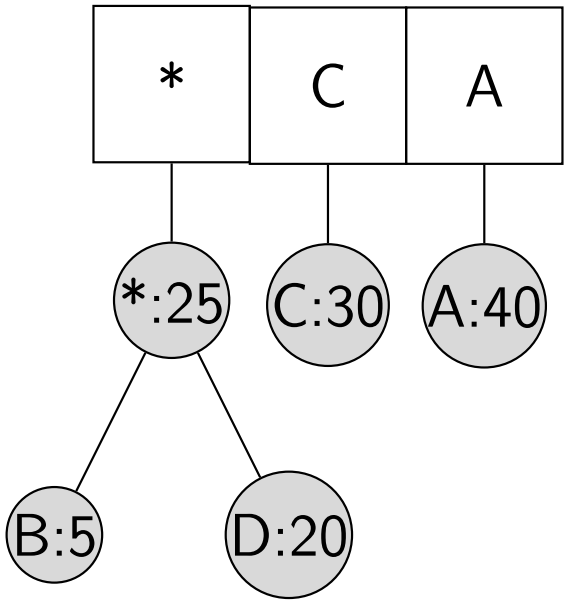
Huffman Oluşturma Örneği

İçinde 4 karakter bulunan ve frekans değerleri aşağıdaki şekilde olan bir metin için Huffman ağacını oluşturalım.

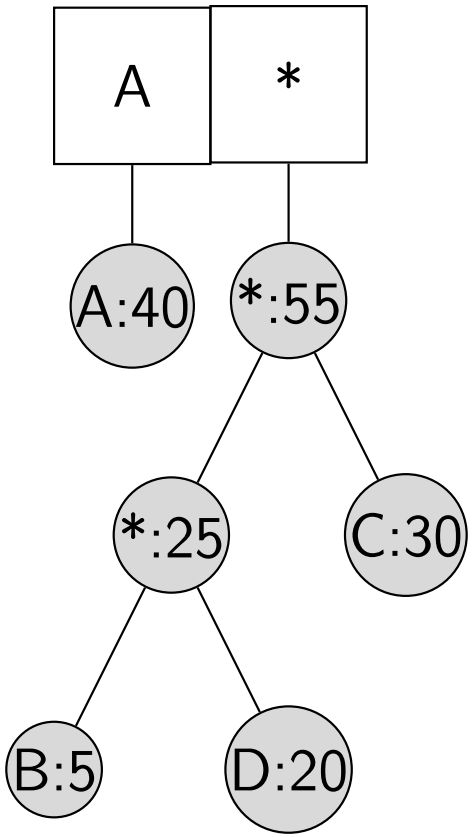
Karakter	**A**	**B**	**C**	**D**
Frekans	40	5	30	20



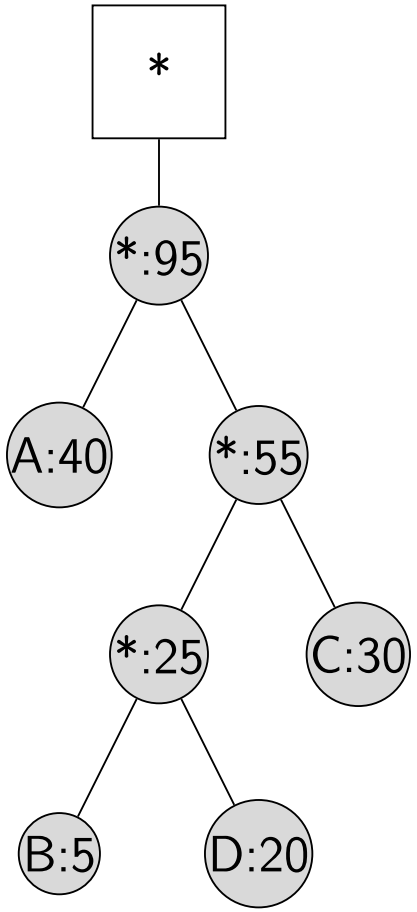
Huffman Oluşturma Örneği



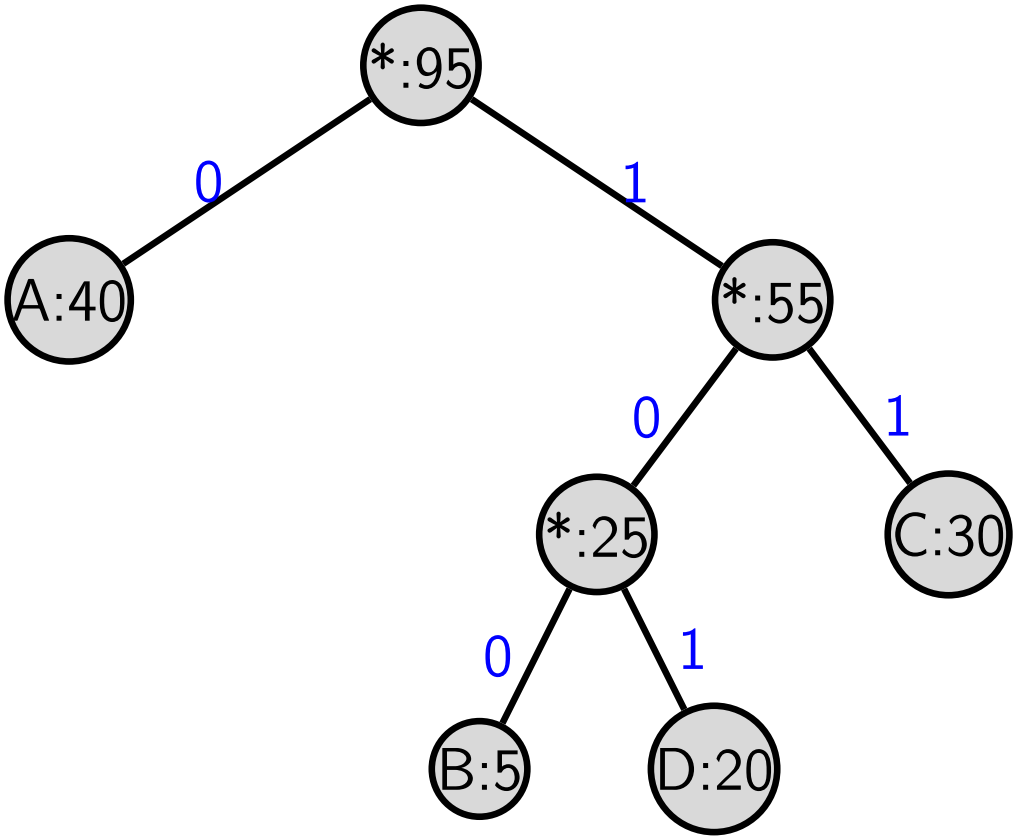
Huffman Oluşturma Örneği



Huffman Oluşturma Örneği



Oluşan Huffman Ağacı



Karakter	**A**	**B**	**C**	**D**
Kod	0	100	11	101

Kodlama ve Kod Çözme Örneği

- **ABACDB** metnini kodlayalım
  - Her karakter 8 bitten oluşursa toplam $6 \cdot 8 = 48$ bit
- A:0 B:100 C:11 D:101

Kodlama ve Kod Çözme Örneği

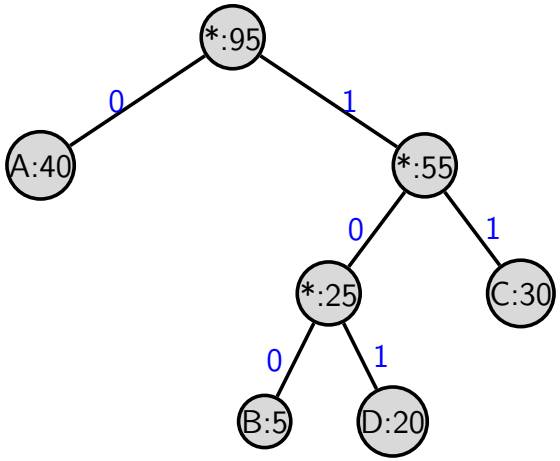
- **ABACDB** metnini kodlayalım
  - Her karakter 8 bitten oluşursa toplam $6 \cdot 8 = 48$ bit
- A:0 B:100 C:11 D:101
- 01000111101100
- 01000111101100 $\rightarrow$ 13 bit

Kodlama ve Kod Çözme Örneği

- Kod çözme için ağacın kökünden başlanır, gelen bitin değerine göre sağa ya da sola yönelinir, yaprağa ulaşınca yapraktaki harf çözülmüş karakteri temsil eder ve tekrar köke dönerek bit dizisinde değer kalmayana kadar devam edilir
  - Eğer kod çözme işlemi yaprakta bitmiyorsa kod veya ağaç hatalıdır

Kodlama ve Kod Çözme Örneği

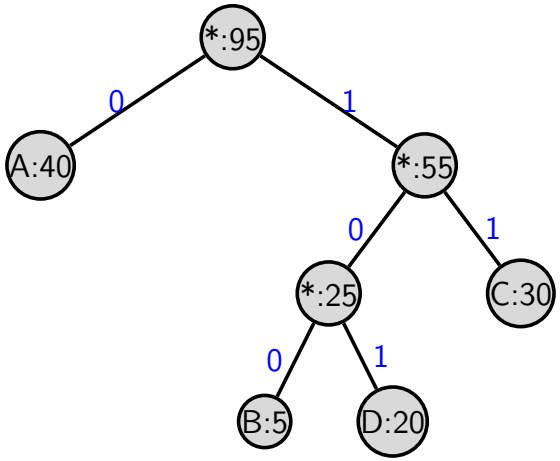
- Kod çözme için ağacın kökünden başlanır, gelen bitin değerine göre sağa ya da sola yönelinir, yaprağa ulaşınca yapraktaki harf çözülmüş karakteri temsil eder ve tekrar köke dönerek bit dizisinde değer kalmayana kadar devam edilir
  - Eğer kod çözme işlemi yaprakta bitmiyorsa kod veya ağaç hatalıdır



- **101011011100101011** kodunu çözelim

Kodlama ve Kod Çözme Örneği

- Kod çözme için ağacın kökünden başlanır, gelen bitin değerine göre sağa ya da sola yönelinir, yaprağa ulaşınca yapraktaki harf çözülmüş karakteri temsil eder ve tekrar köke dönerek bit dizisinde değer kalmayana kadar devam edilir
  - Eğer kod çözme işlemi yaprakta bitmiyorsa kod veya ağaç hatalıdır



- **10101101110010101011** kodunu çözelim

D	A	C	A	C	B	D	A	C
101	0	11	0	11	100	101	0	11

- 10101101110010101011

Soru

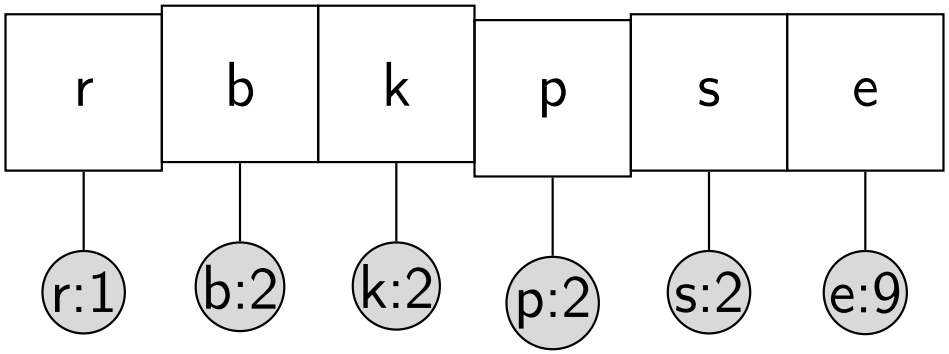
- **beekeeperkeepsbees** metnindeki harflerin frekansını bulun ve Hufmann kodlama ile kodlayın.
- Frekansı aynı olan düğümler için:
  - Ağaçlarda değer kısmı(harf) en küçük olan düğümler bulunur
  - İki ağaçtaki en küçük değerlerden küçük olanı solda yer alır

Soru

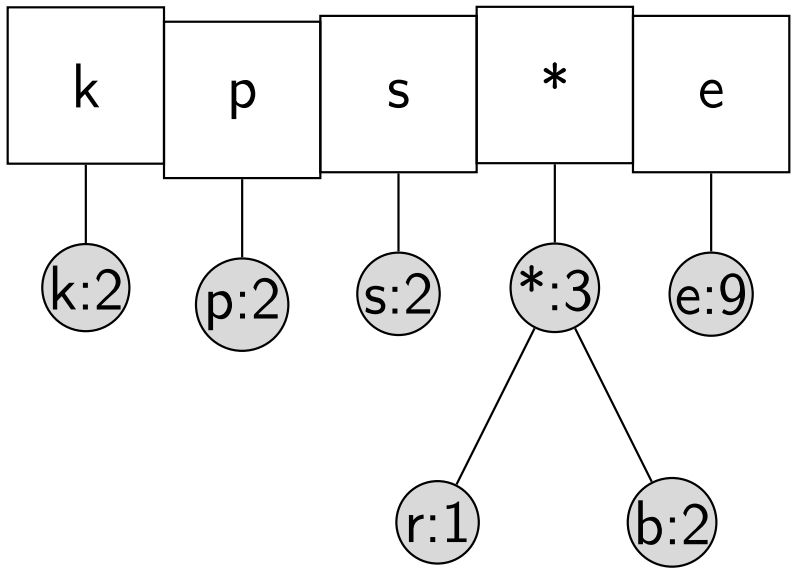
- **beekeeperkeepsbees** metnindeki harflerin frekansını bulun ve Hufmann kodlama ile kodlayın.
- Frekansı aynı olan düğümler için:
  - Ağaçlarda değer kısmı(harf) en küçük olan düğümler bulunur
  - İki ağaçtaki en küçük değerlerden küçük olanı solda yer alır
- Frekanslar $\rightarrow$ **r**:1 **b**:2 **k**:2 **p**:2 **s**:2 **e**:9

Soru

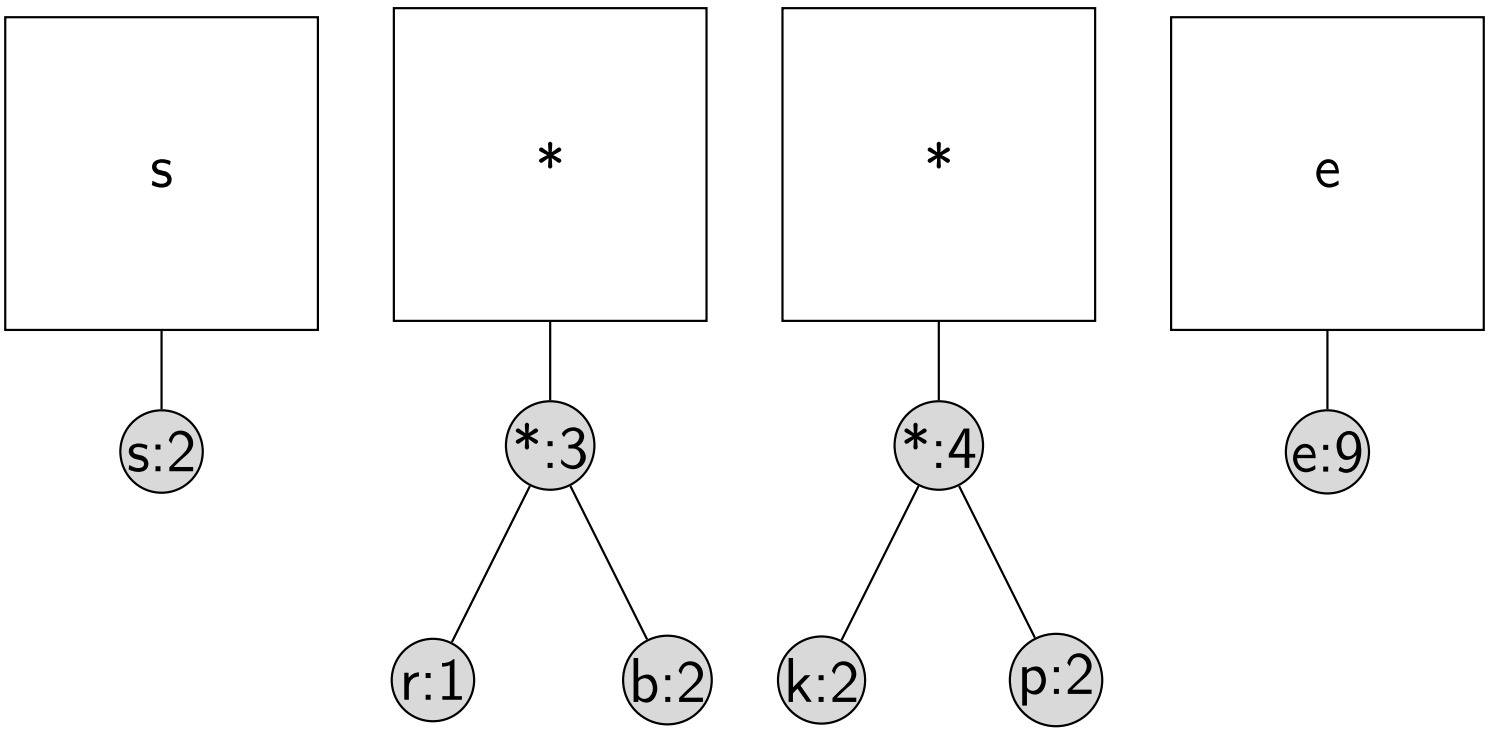
- **beekeeperkeepsbees** metnindeki harflerin frekansını bulun ve Hufmann kodlama ile kodlayın.
- Frekansı aynı olan düğümler için:
  - Ağaçlarda değer kısmı(harf) en küçük olan düğümler bulunur
  - İki ağaçtaki en küçük değerlerden küçük olanı solda yer alır
- Frekanslar $\rightarrow$ **r**:1 **b**:2 **k**:2 **p**:2 **s**:2 **e**:9



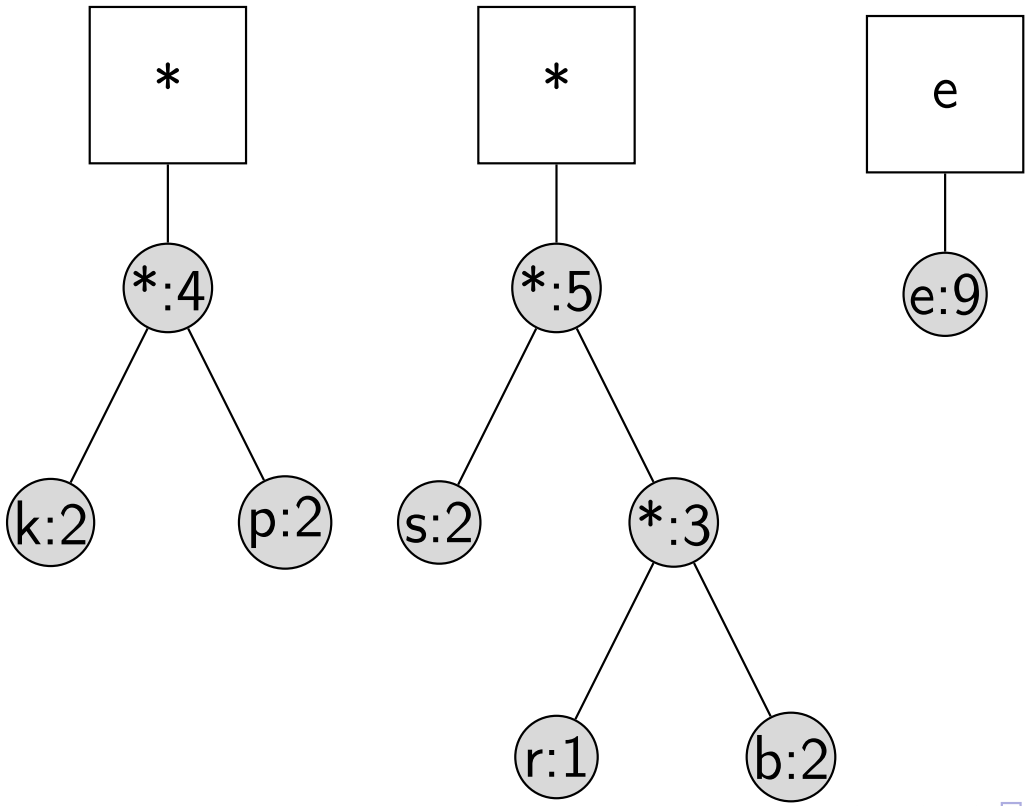
Çözüm



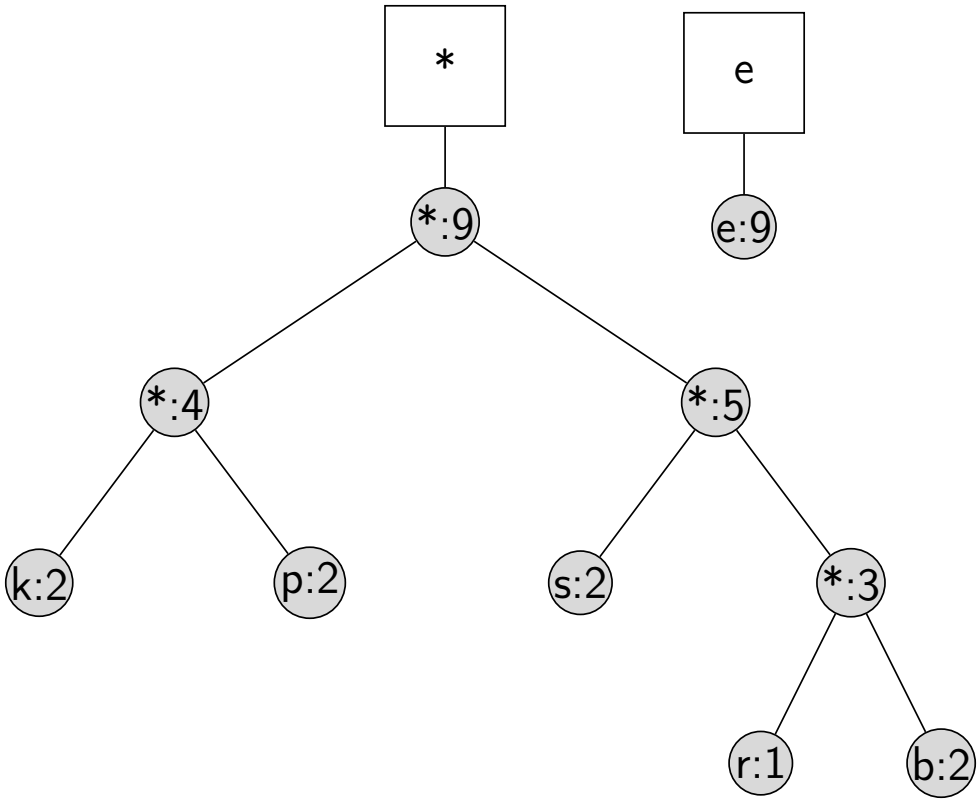
Çözüm



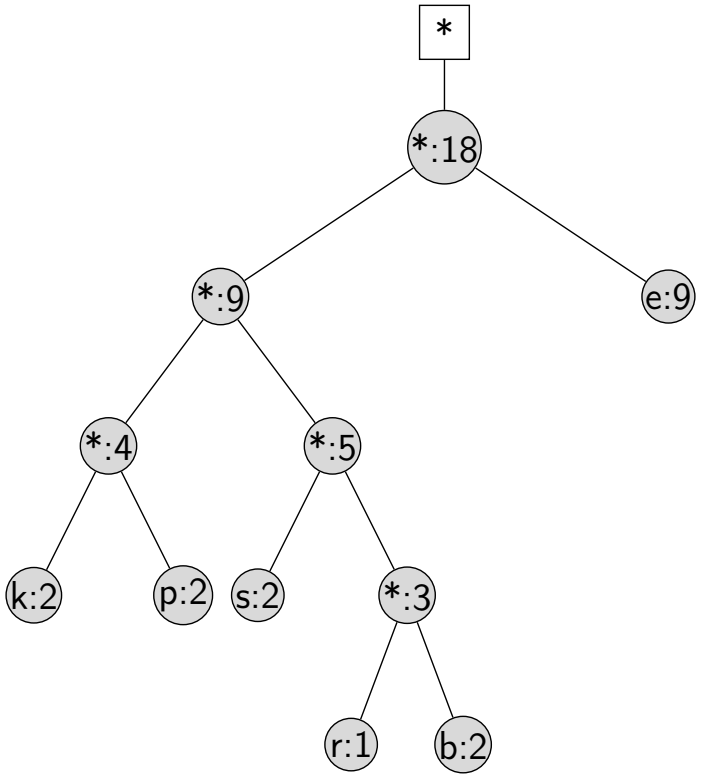
Çözüm



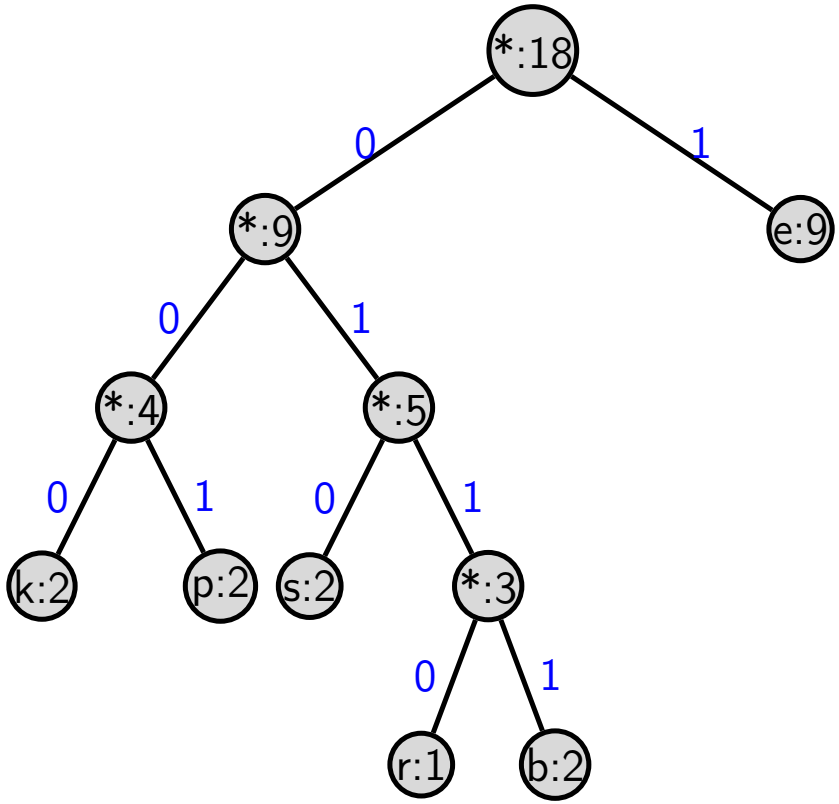
Çözüm



Çözüm

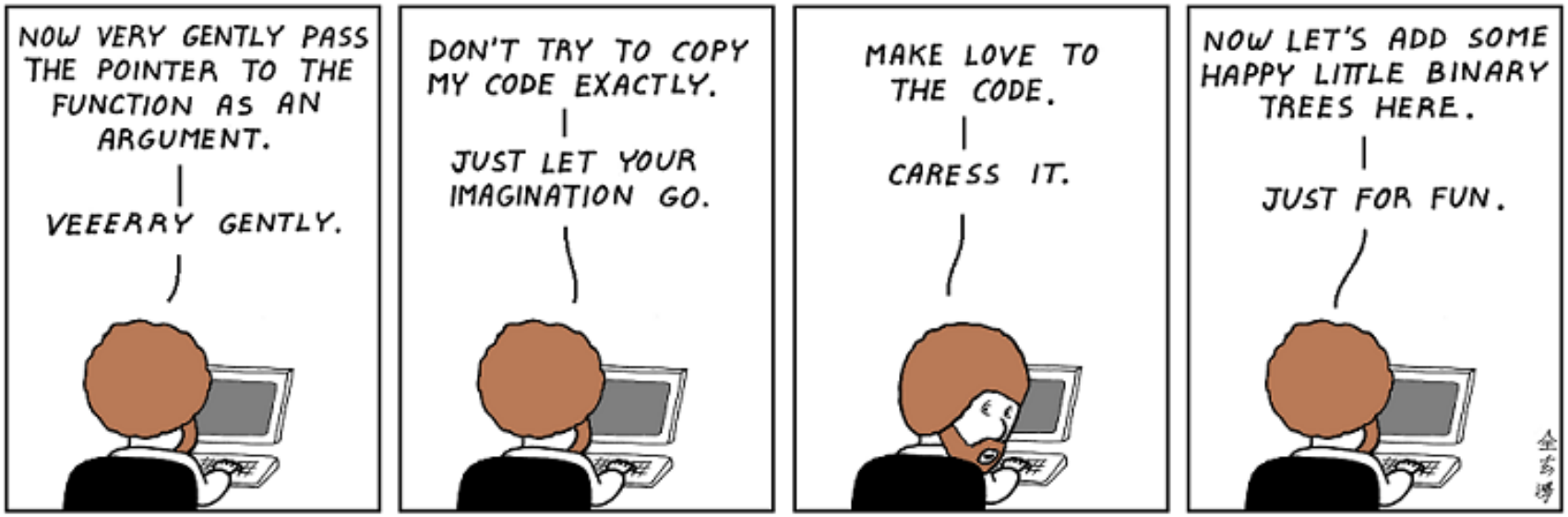


Çözüm



Harf	r	b	k	p	s	e
Kod	0110	0111	000	001	010	1

- Kodlanmış metin: **011111000110011011000011001010011111010**
- Bit sayısı: **39**, Sıkıştırma oranı: $\frac{39}{18 \cdot 8} \approx \%27$



The Joy of Programming
with Bob Ross

1

[1] abstrusegoose