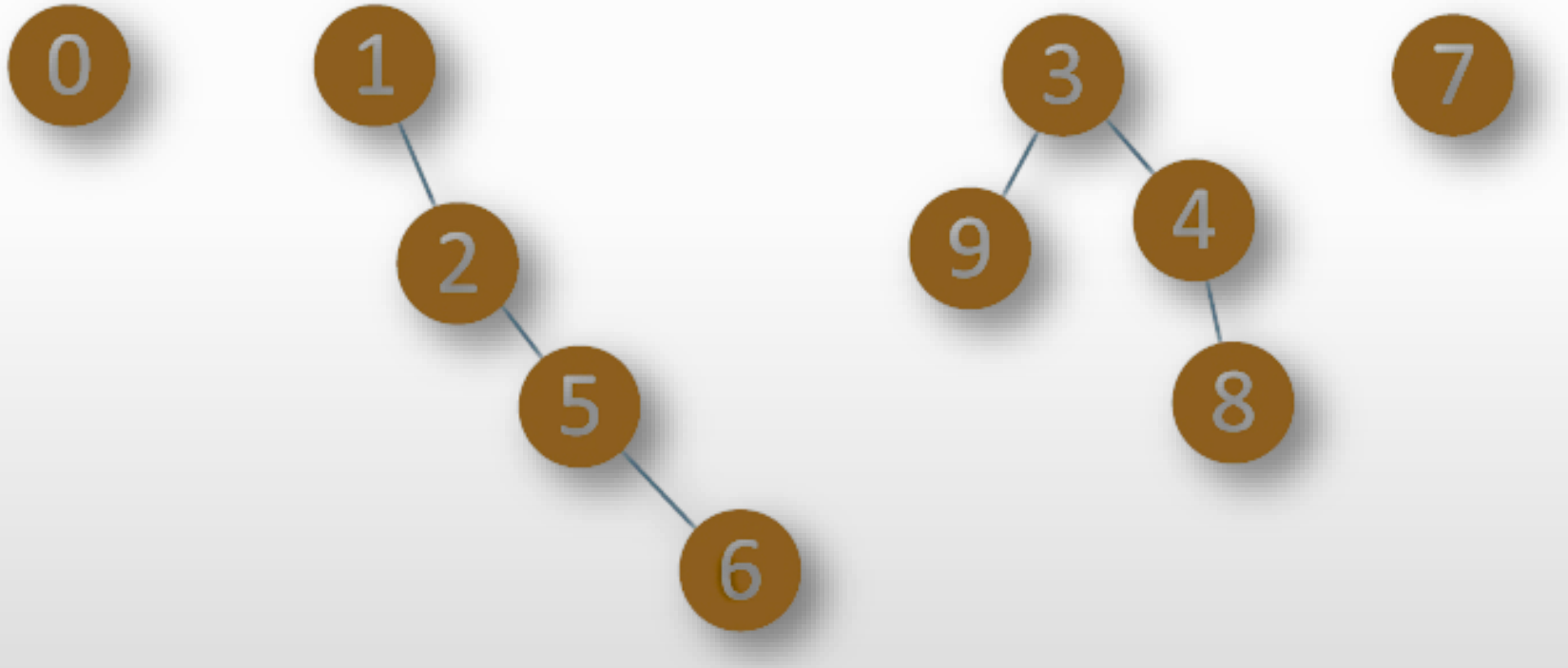
Bölüm 12: Kümeler

Veri Yapıları

Kümeler



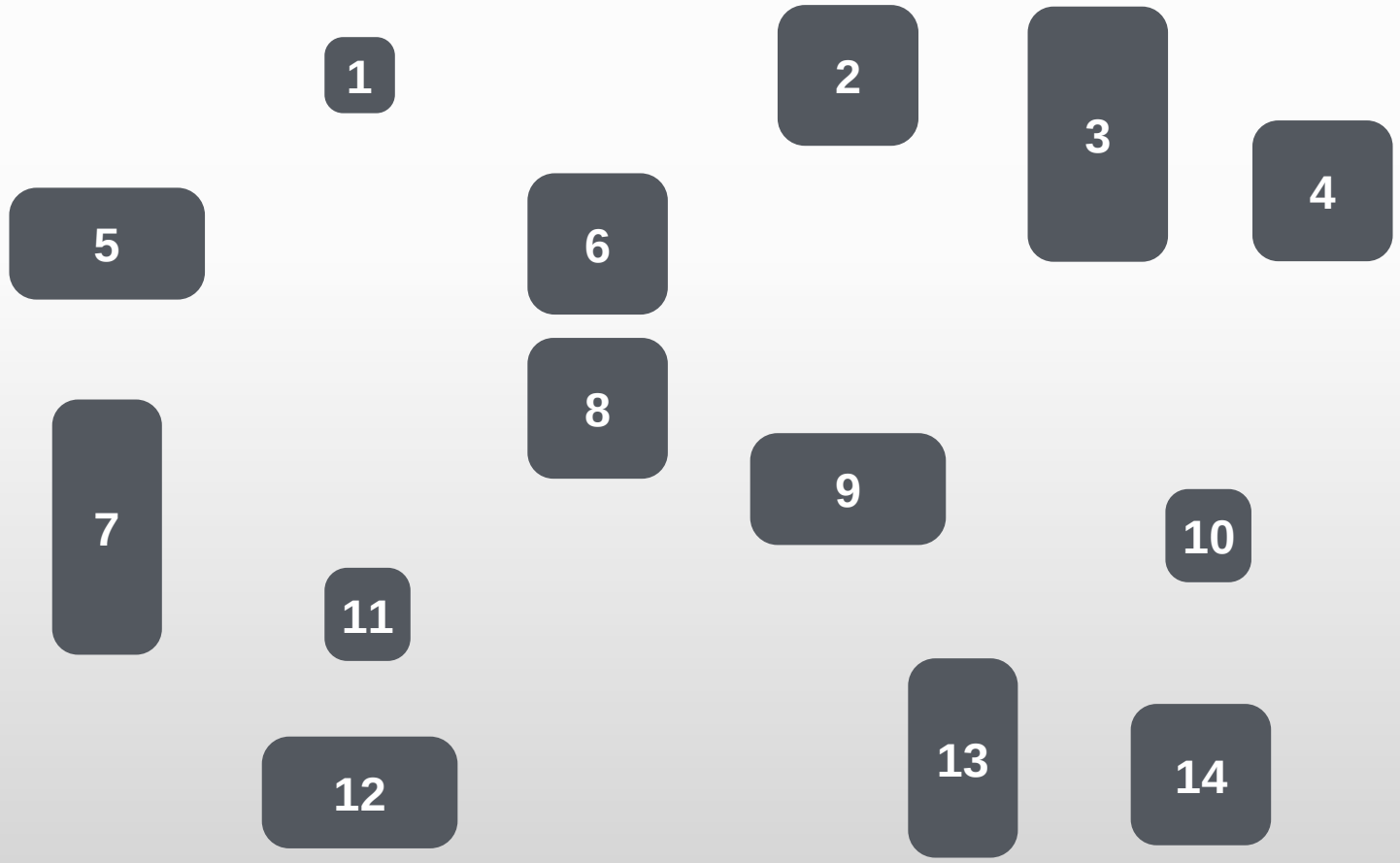
Kümeler

- Benzersiz (tekil) öğelerden oluşur.
- Çizge teorisinde, ağ algoritmalarında ve benzeri alanlarda kullanılır.
- Union-Find, bir veya daha fazla ayrık kümeden oluşan veri yapısıdır.
- Temel olarak iki ana işlemi vardır:
  - find (bul): Bir elemanın ait olduğu kümeyi bulma
  - union (birleştir): İki kümeyi birleştirme.

Mıknatıs Örneği

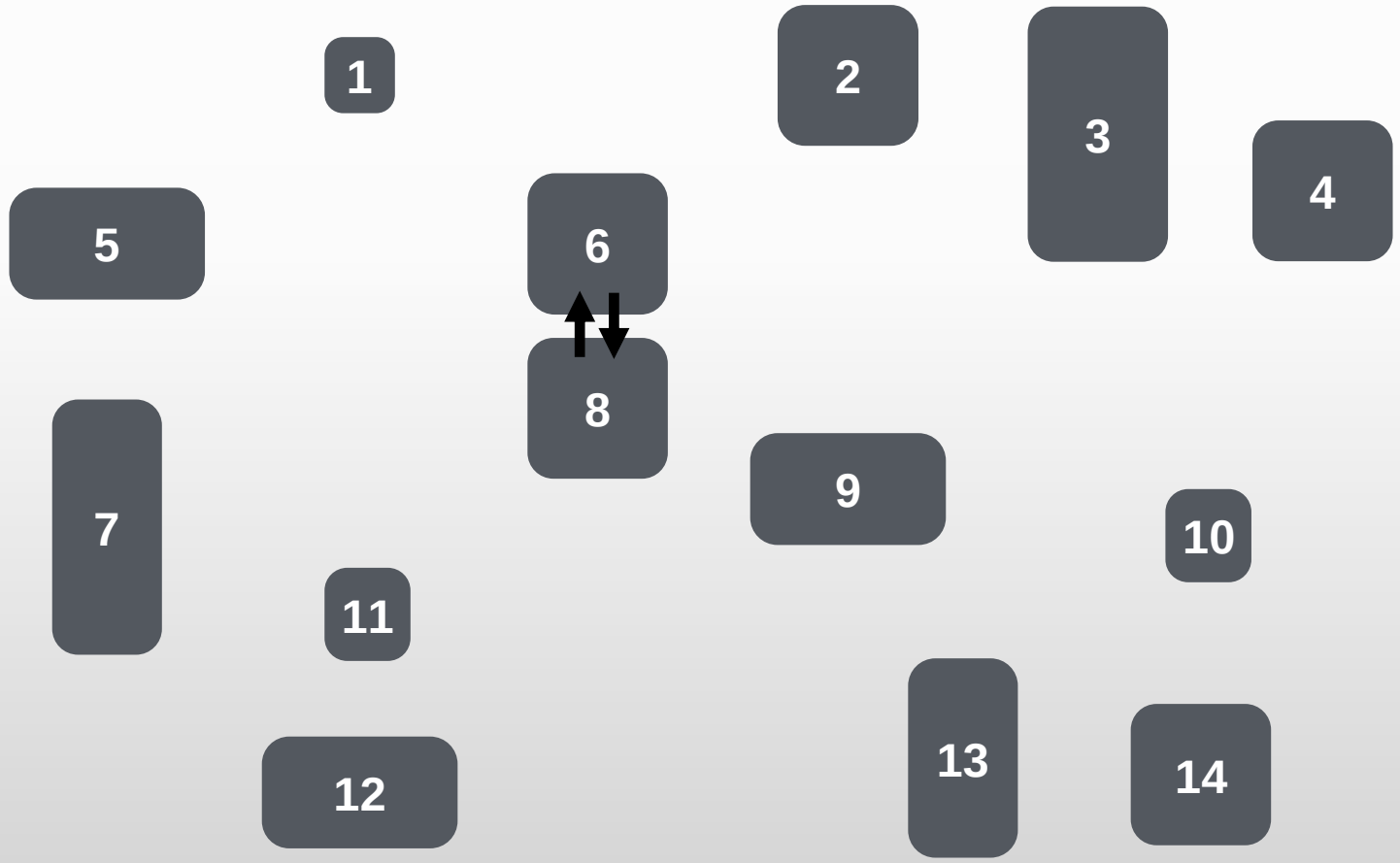
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



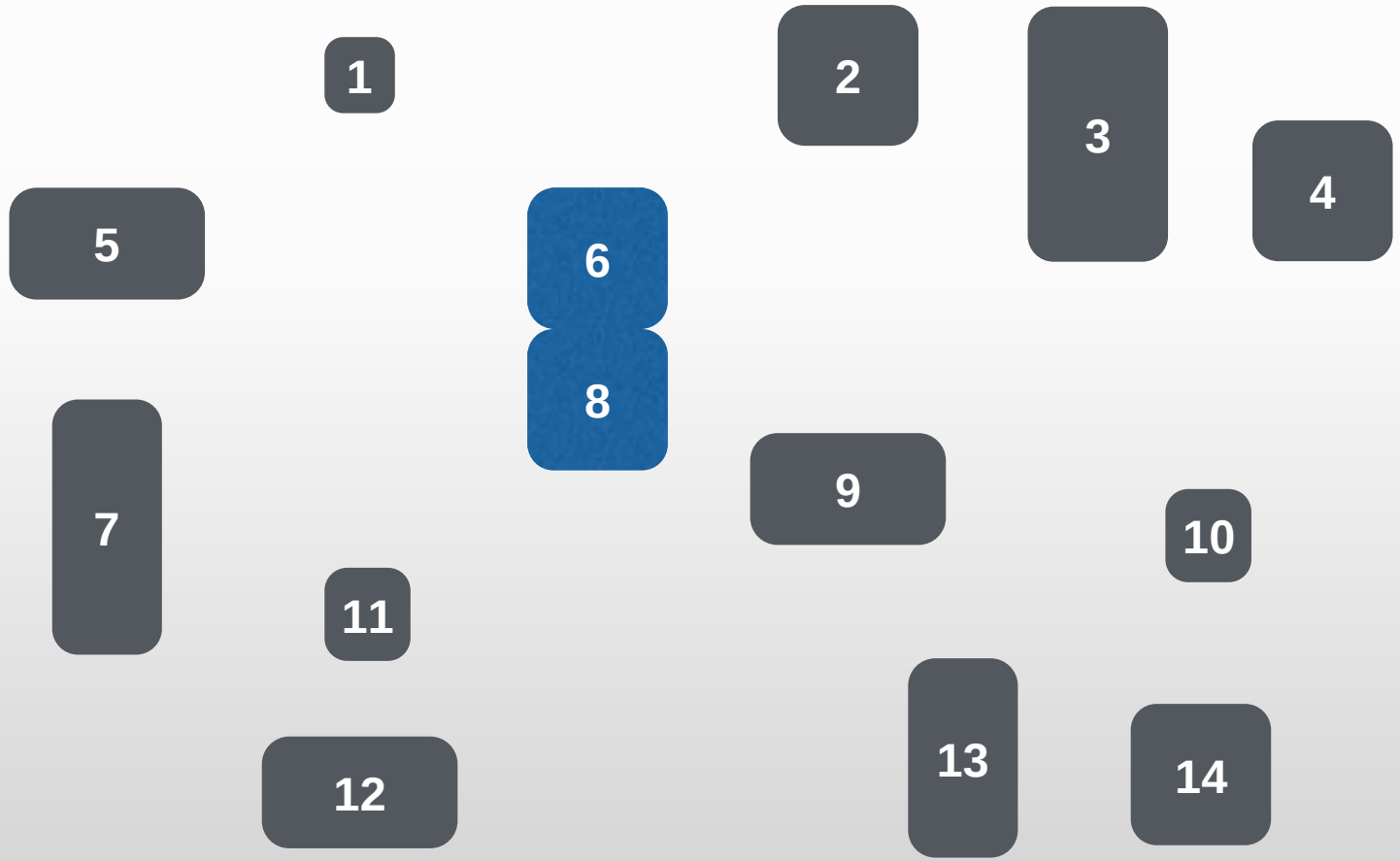
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



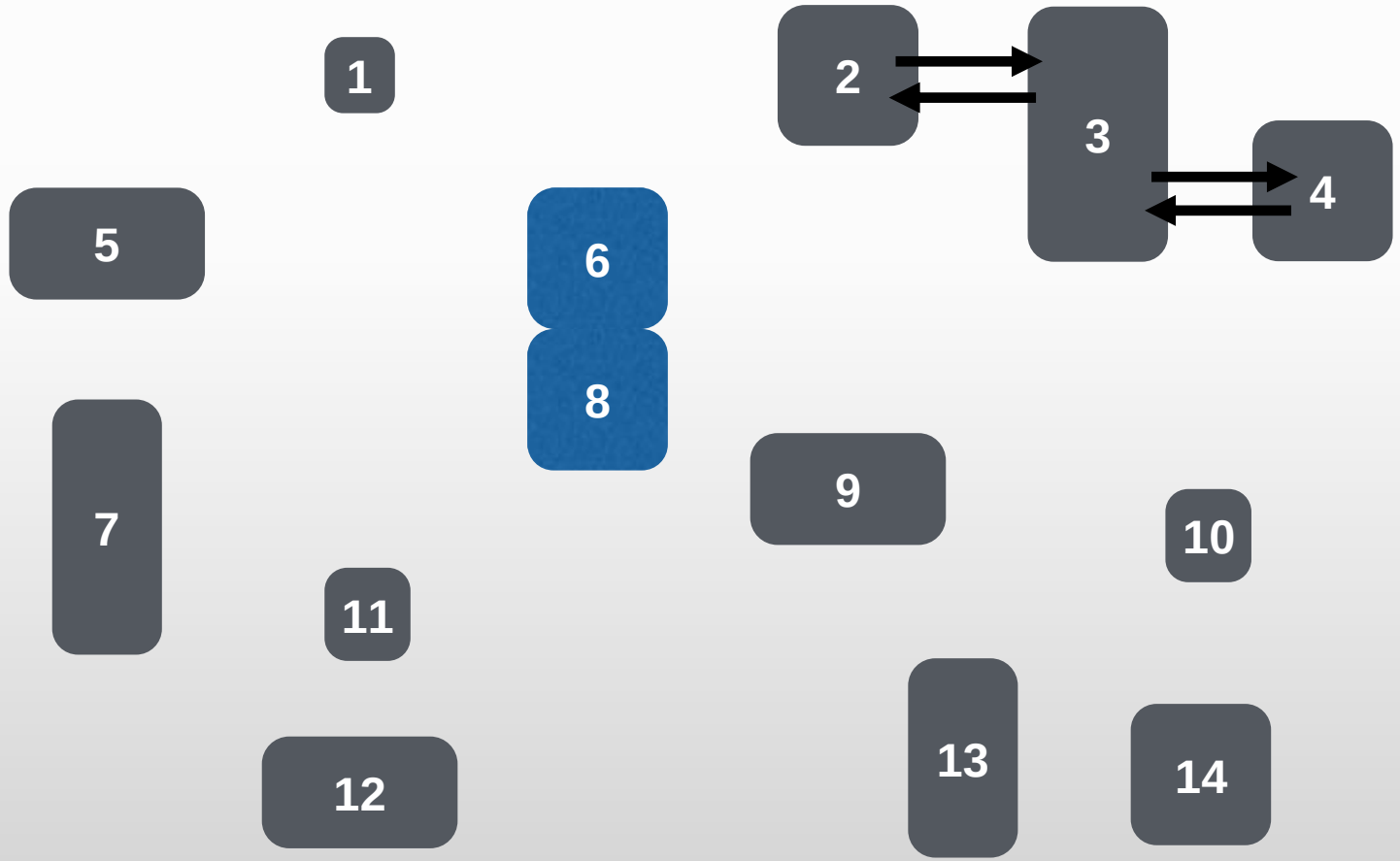
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



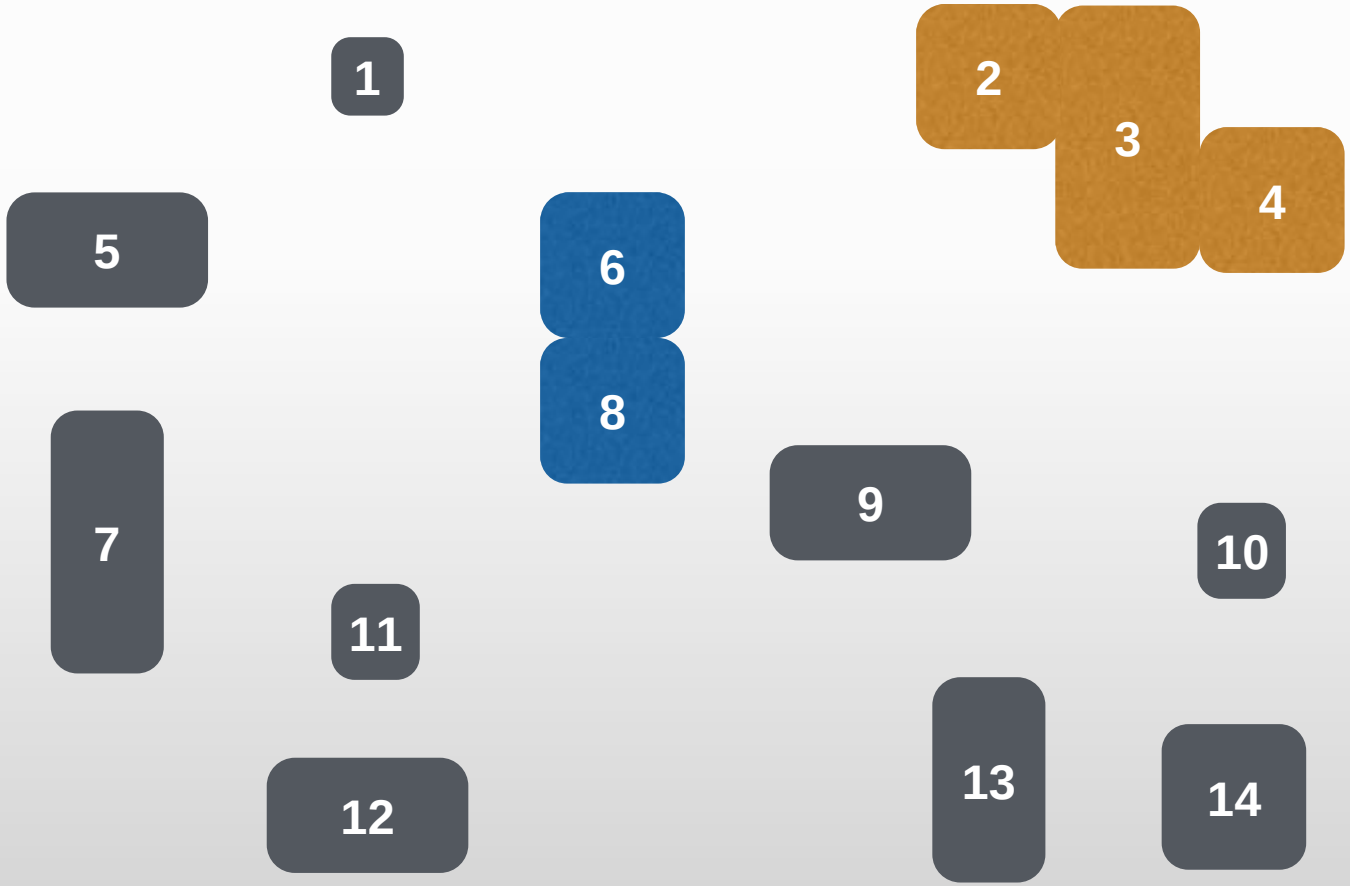
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



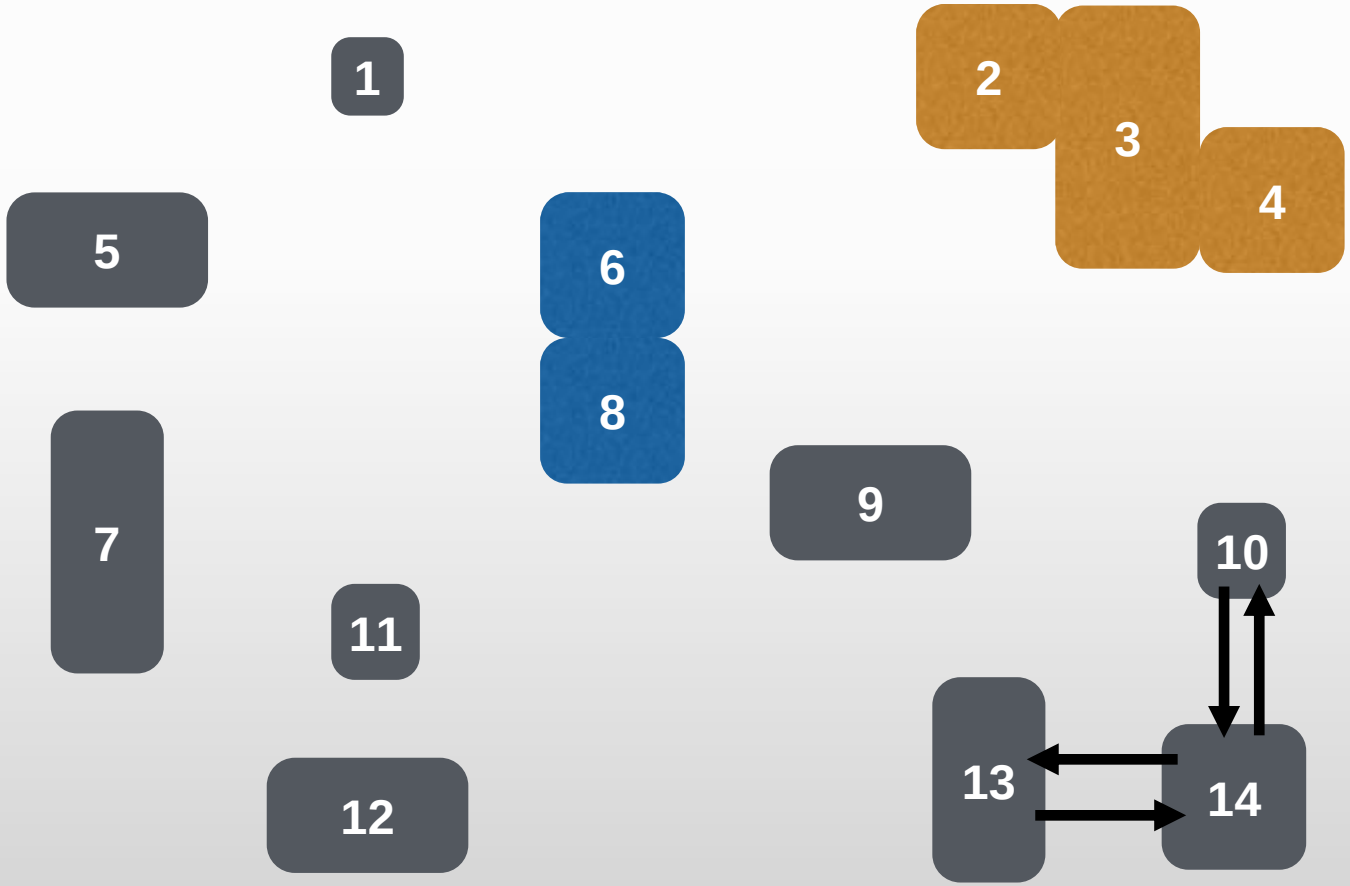
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



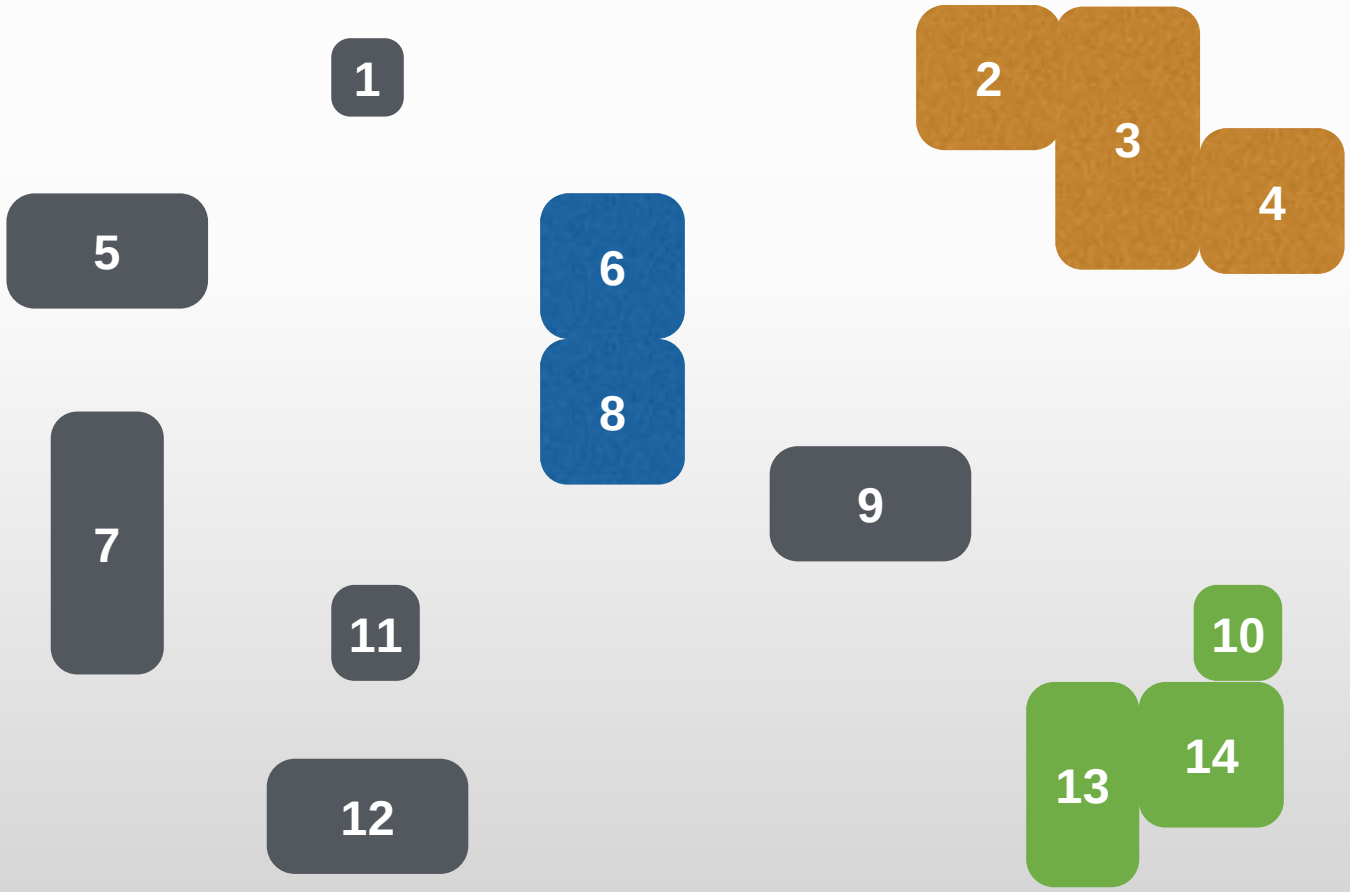
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



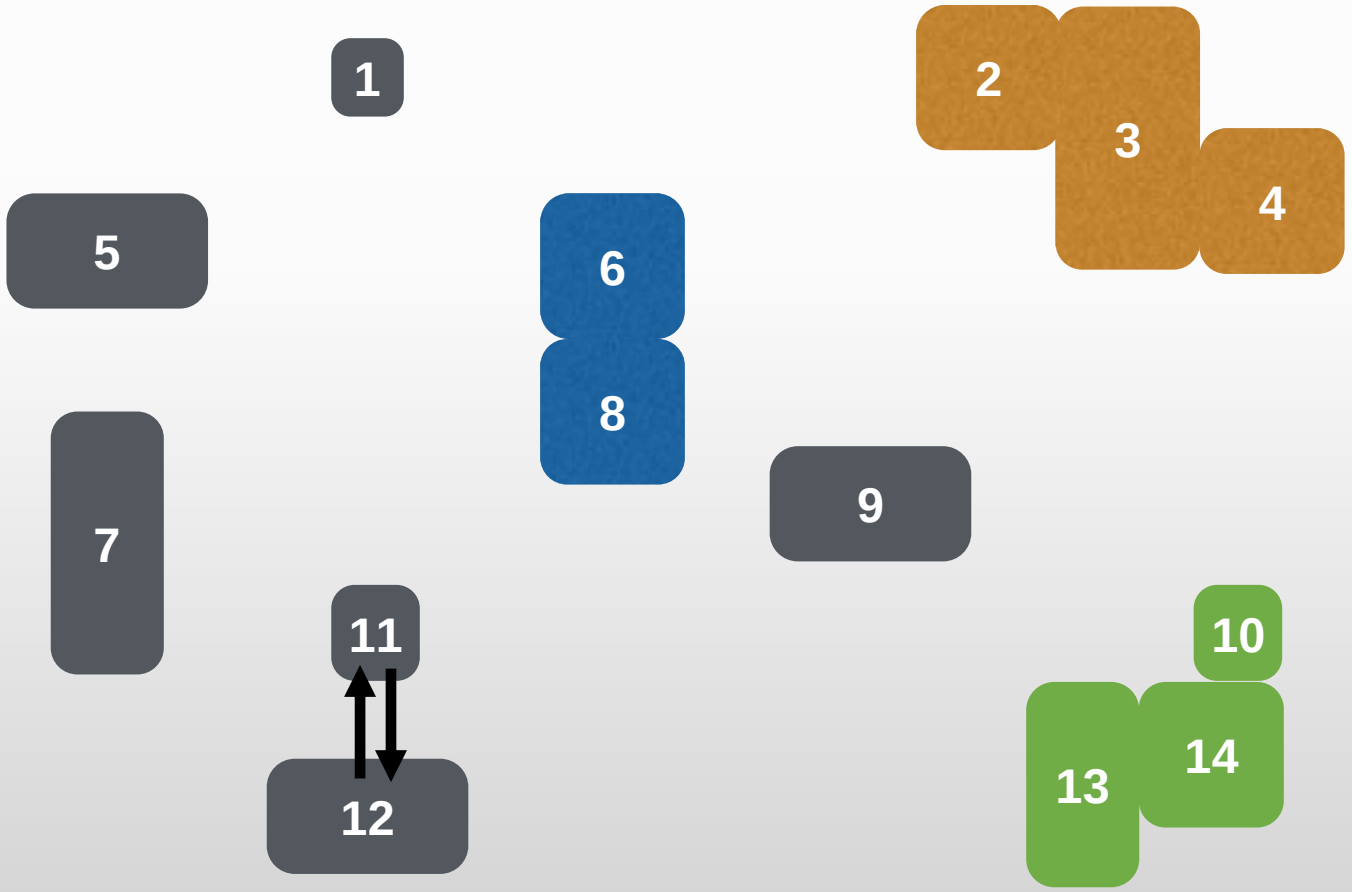
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



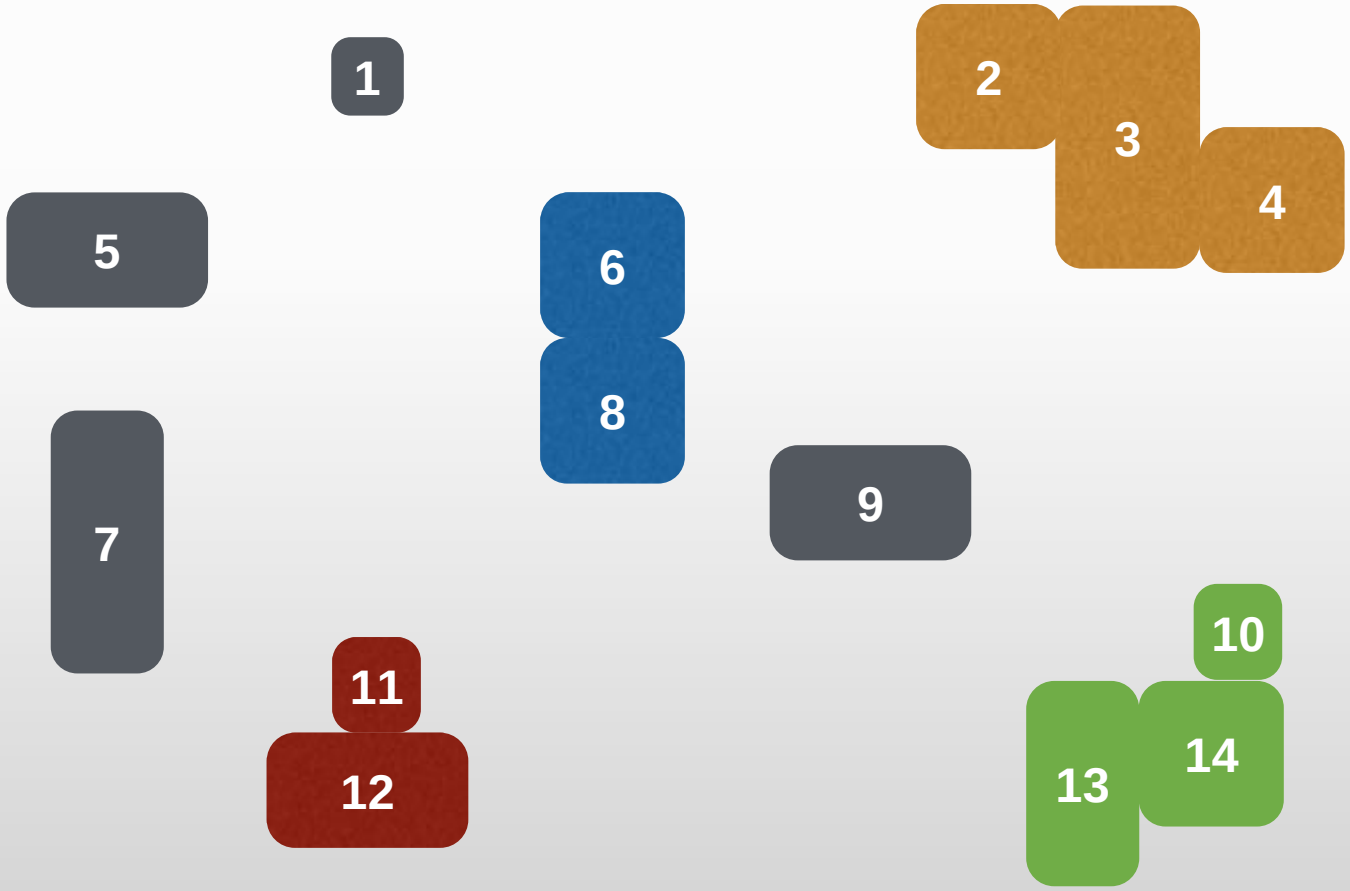
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



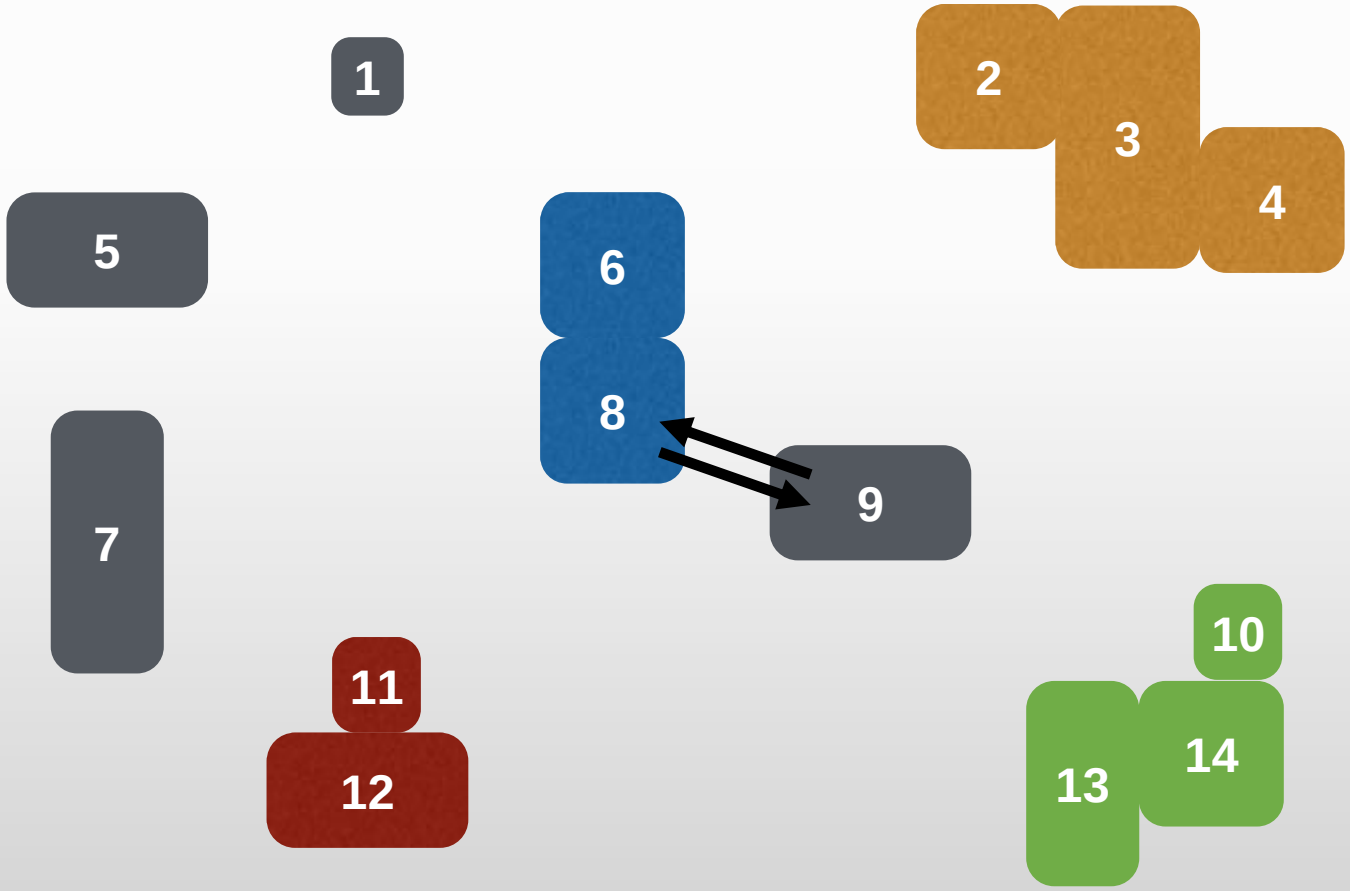
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



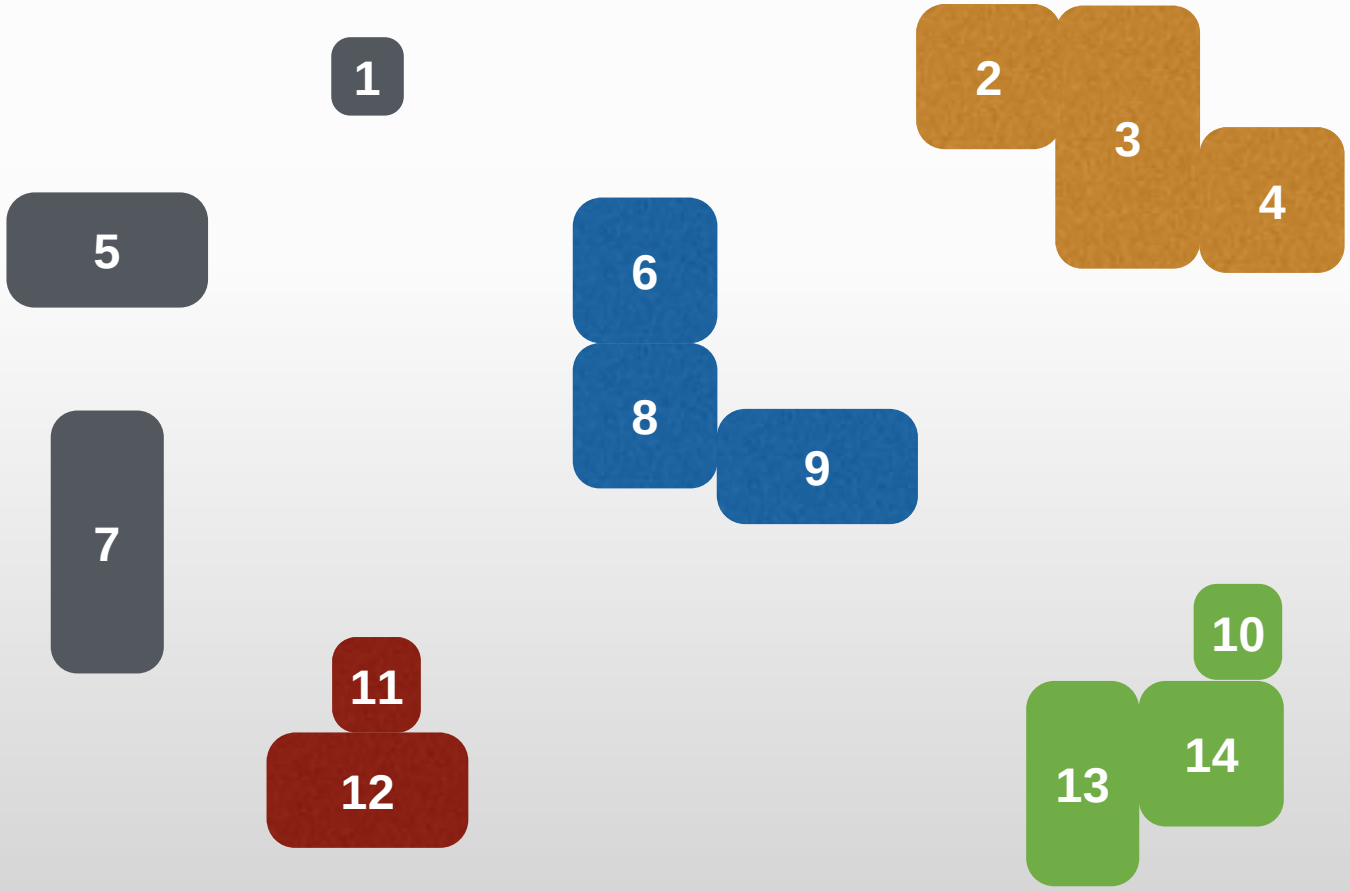
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



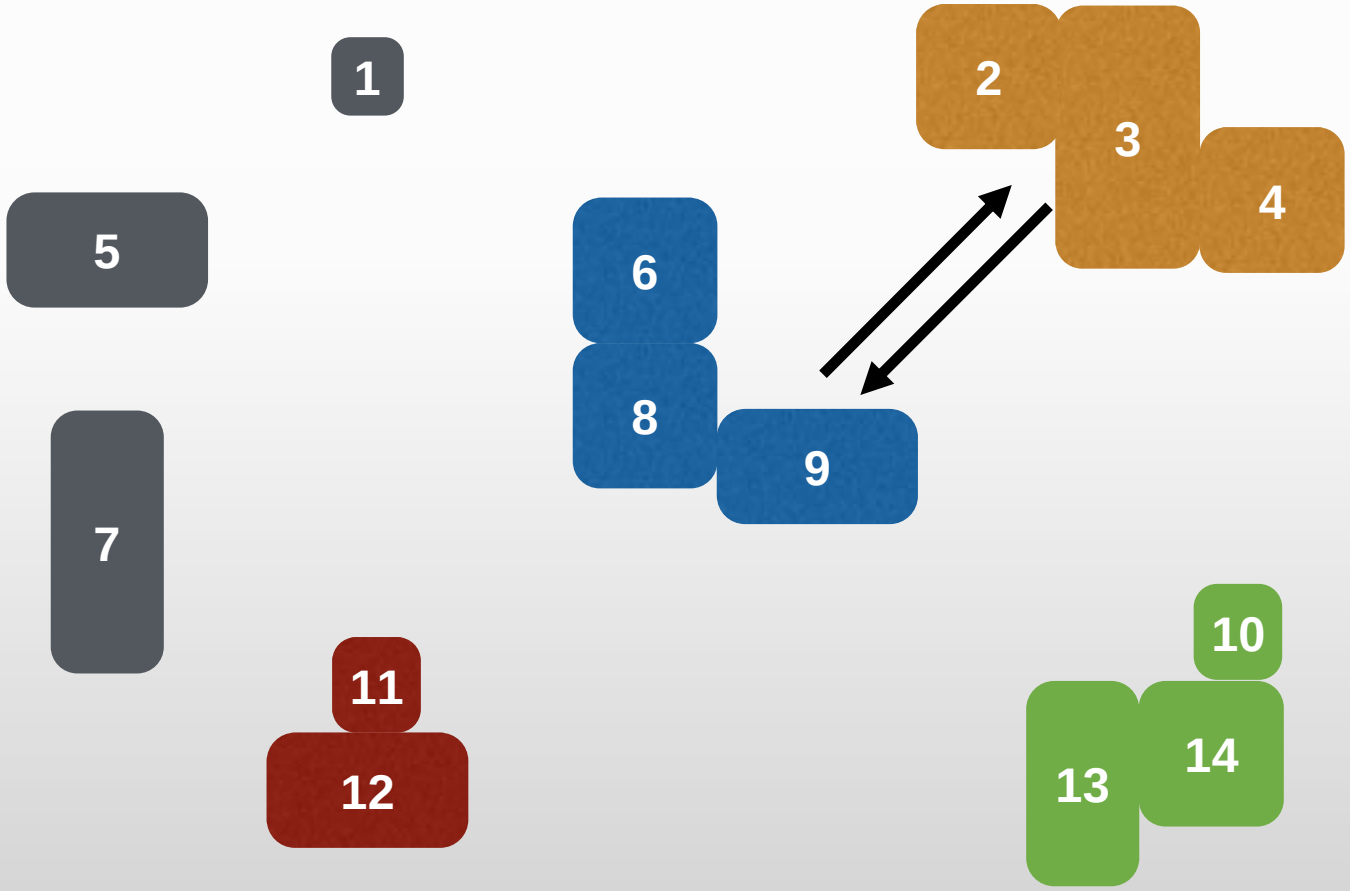
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



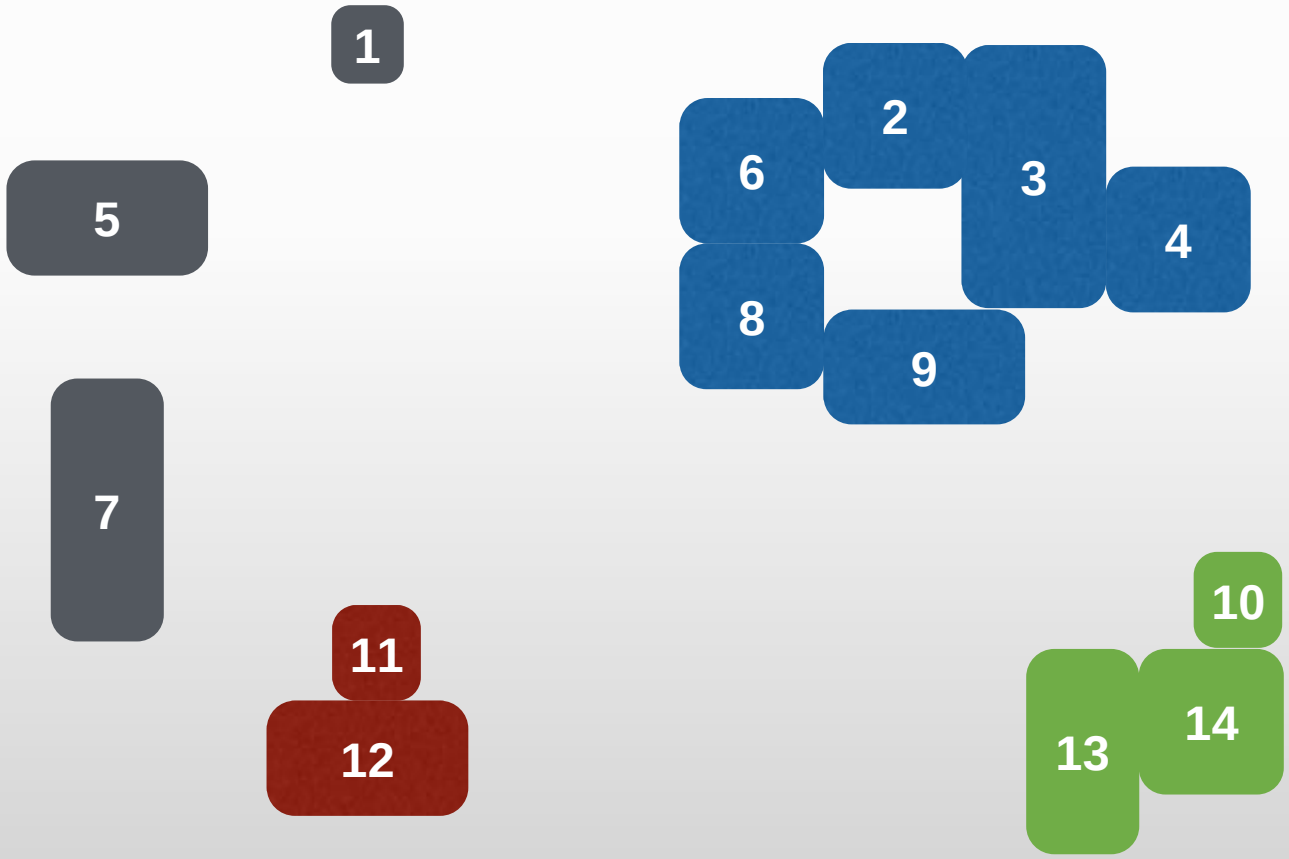
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



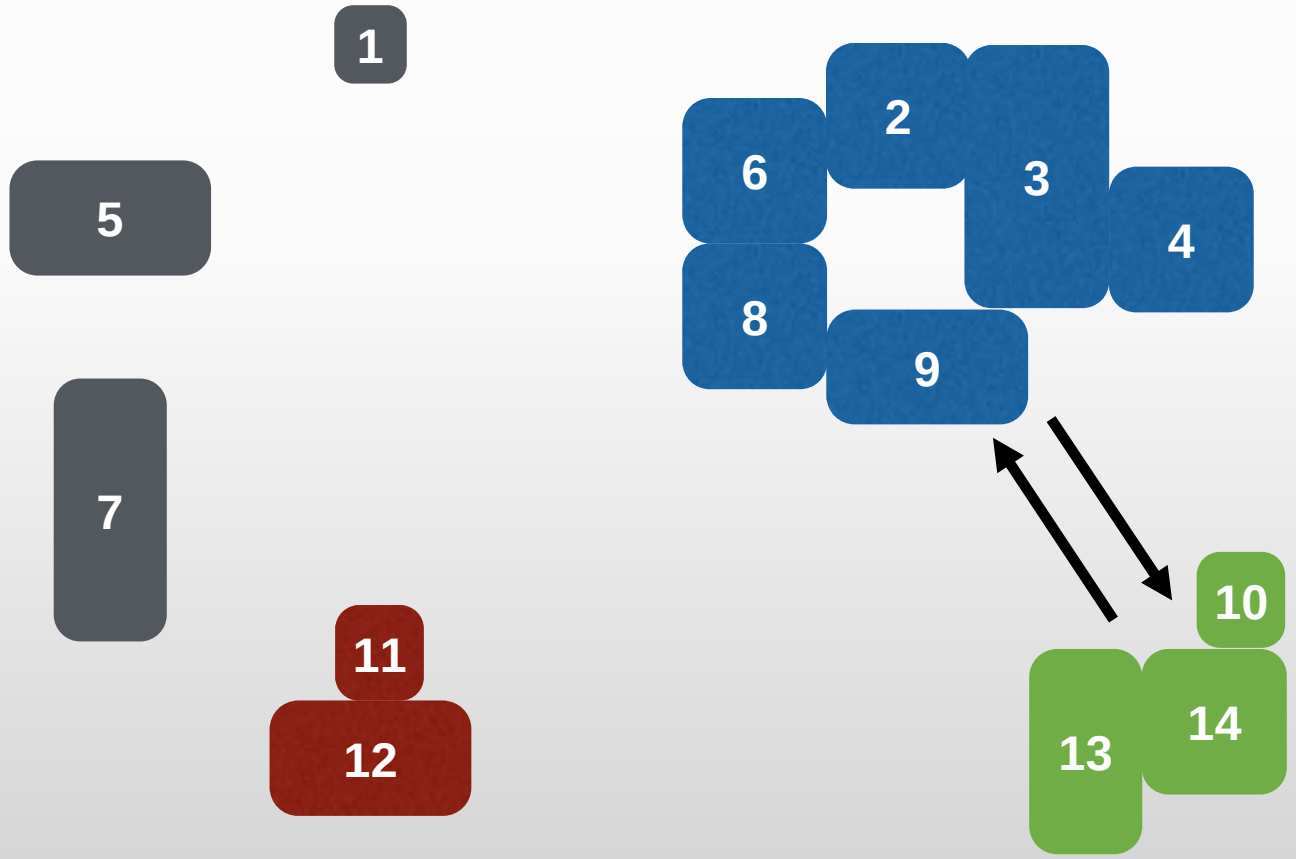
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



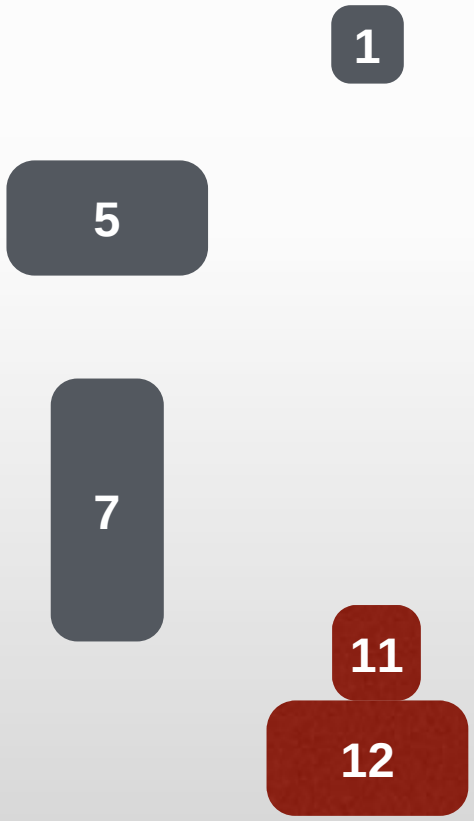
Mıknatıs Örneği

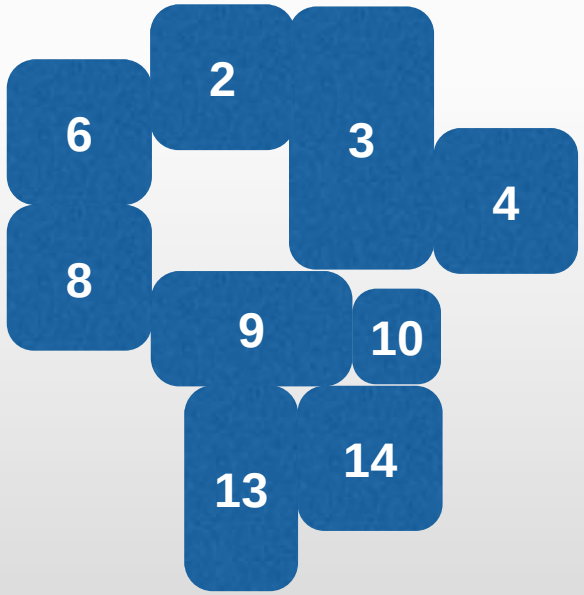
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



Mıknatıs Örneği

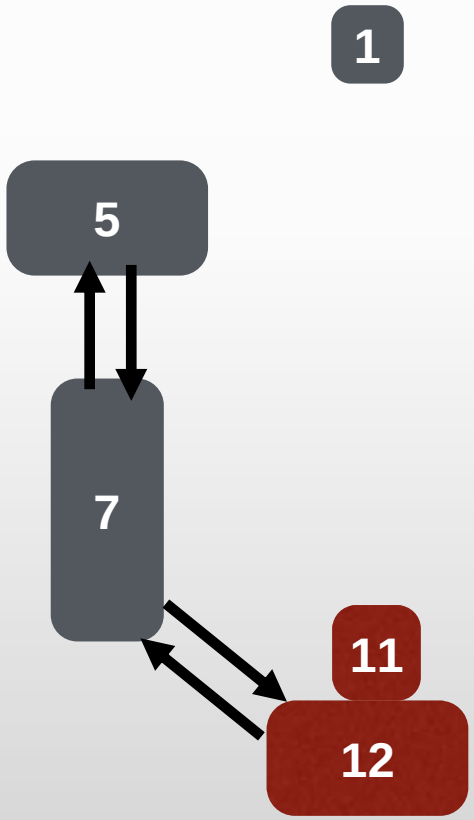
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14

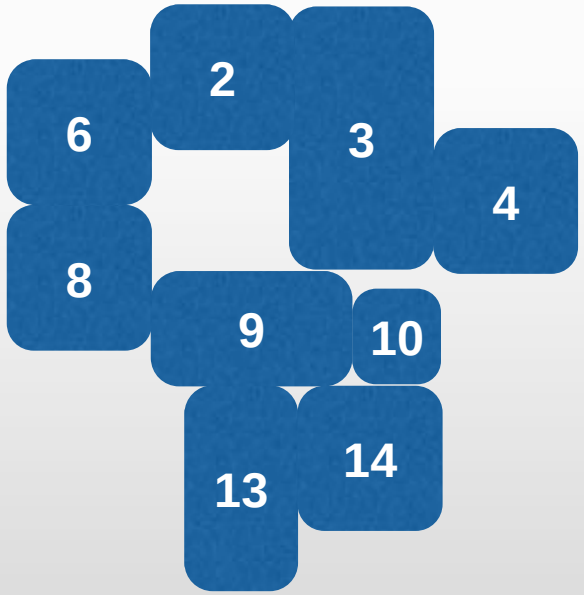




Mıknatıs Örneği

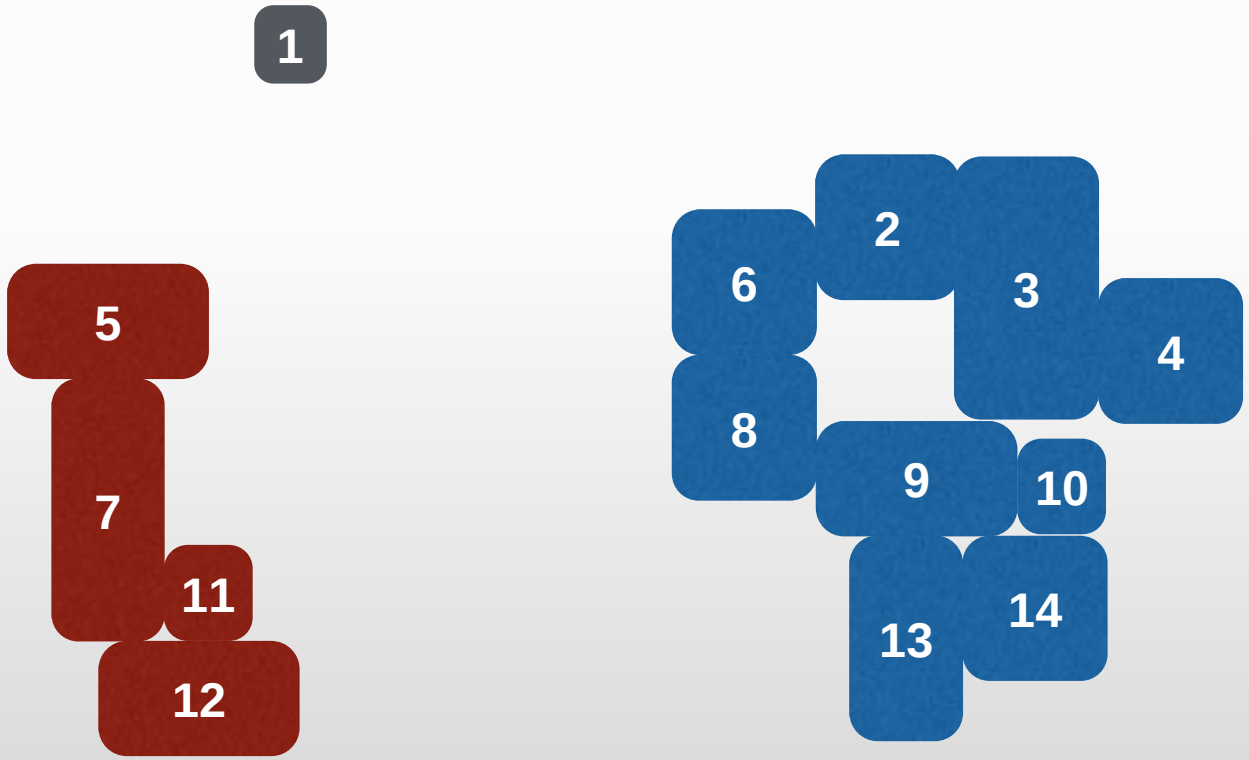
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14





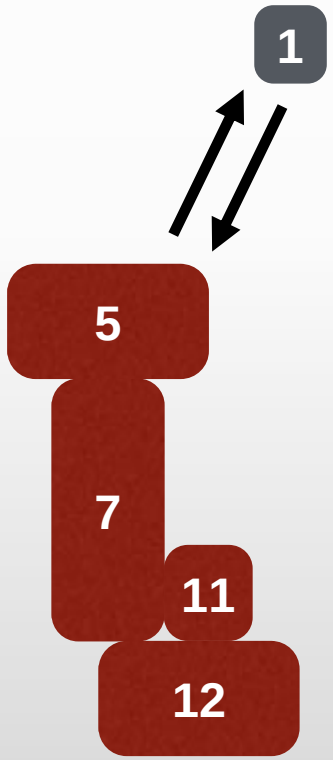
Mıknatıs Örneği

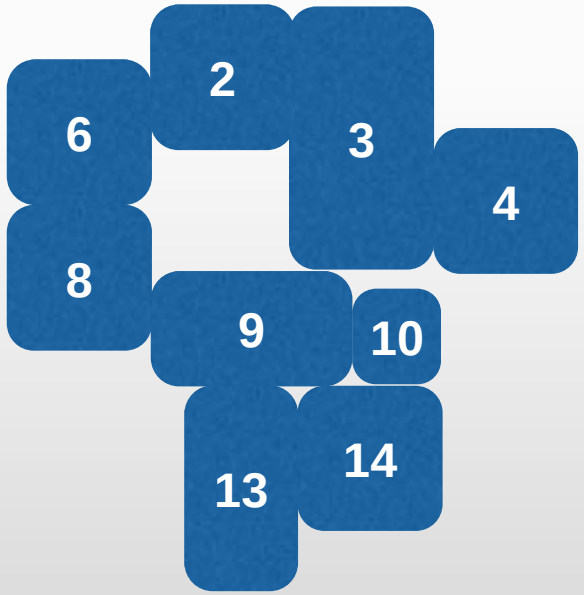
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



Mıknatıs Örneği

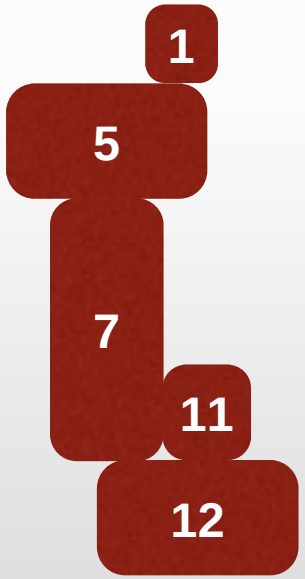
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14

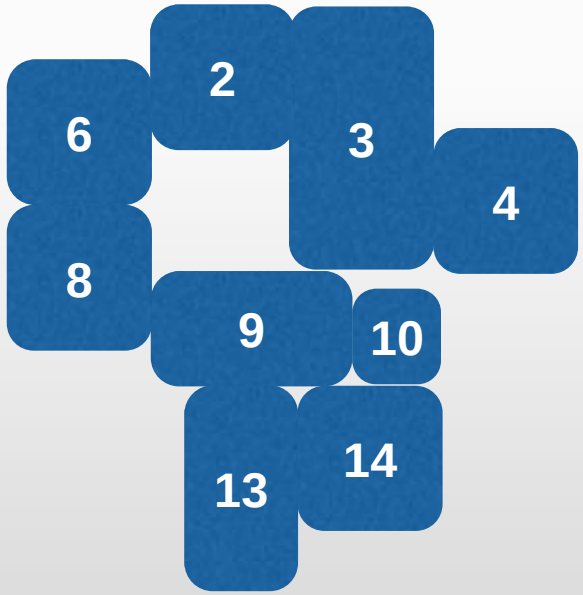




Mıknatıs Örneği

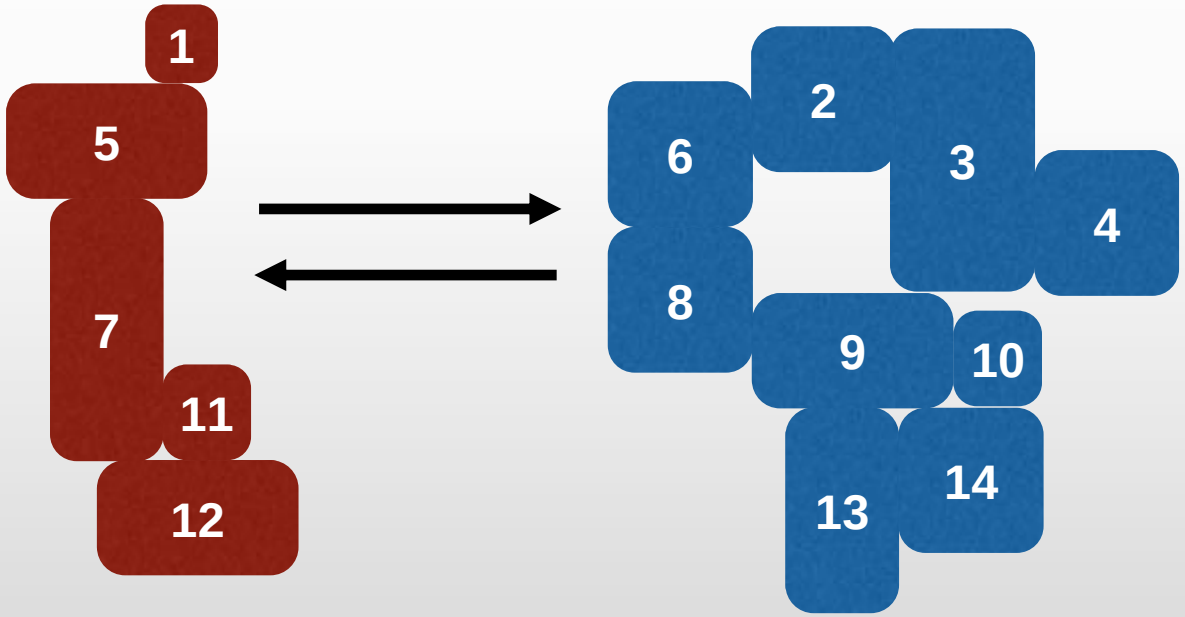
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14





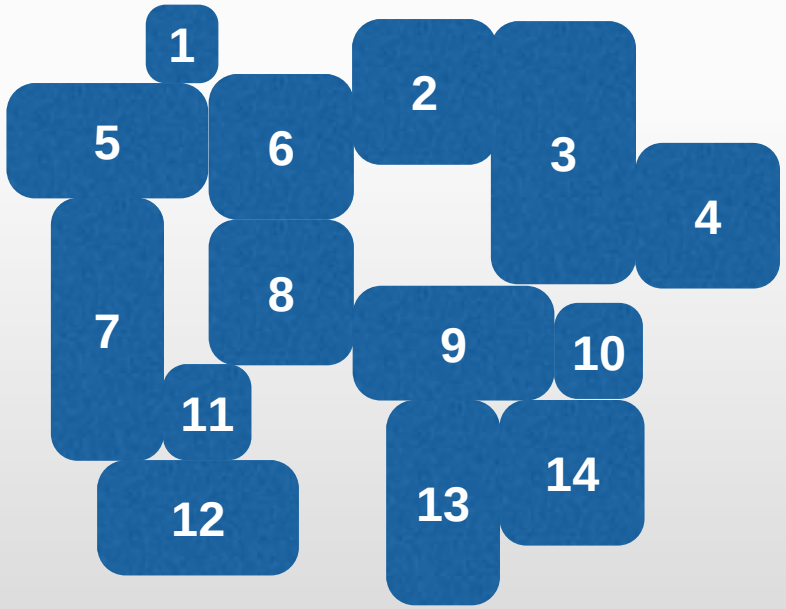
Mıknatıs Örneği

Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



Mıknatıs Örneği

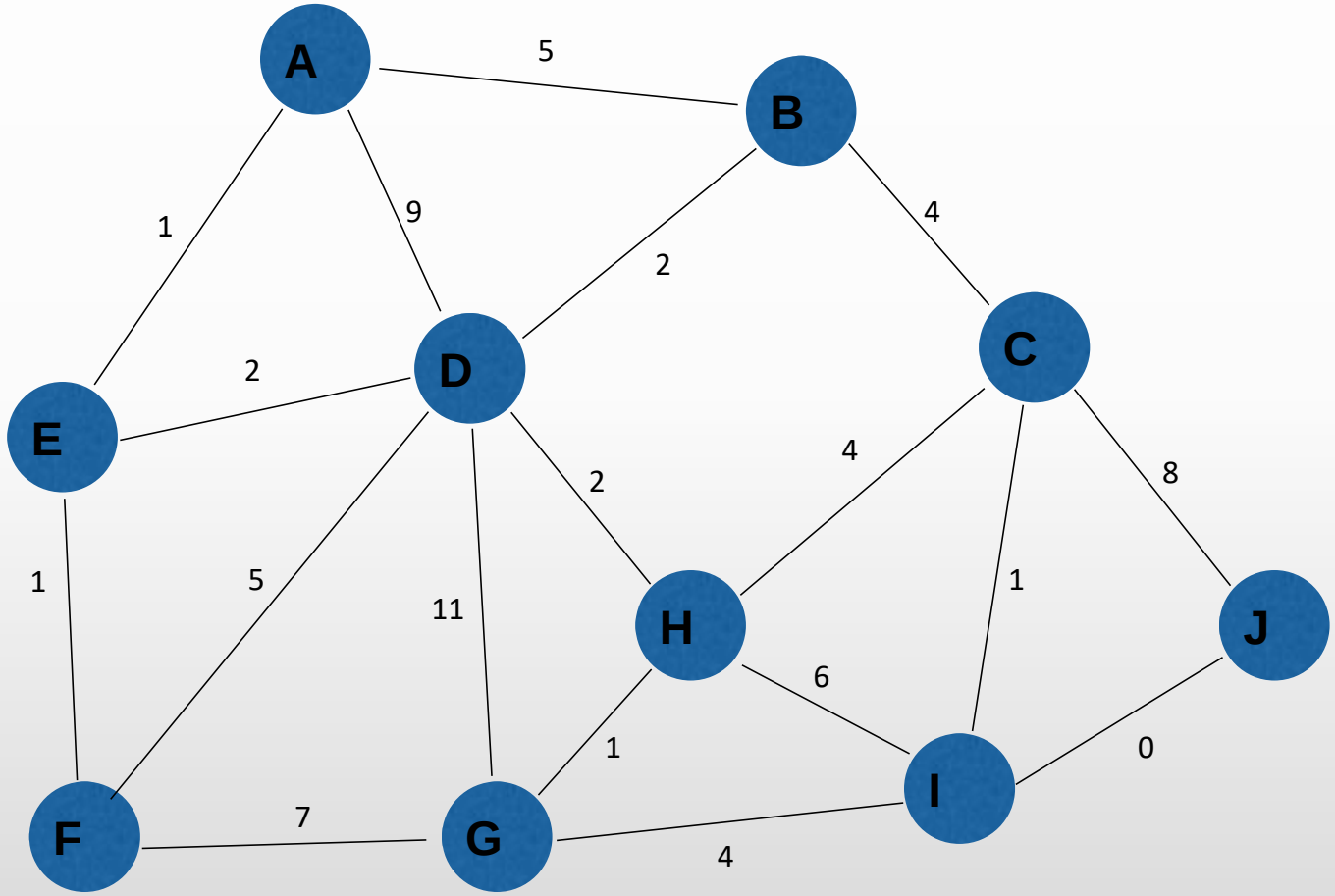
Magnet 1
Magnet 2
Magnet 3
Magnet 4
Magnet 5
Magnet 6
Magnet 7
Magnet 8
Magnet 9
Magnet 10
Magnet 11
Magnet 12
Magnet 13
Magnet 14



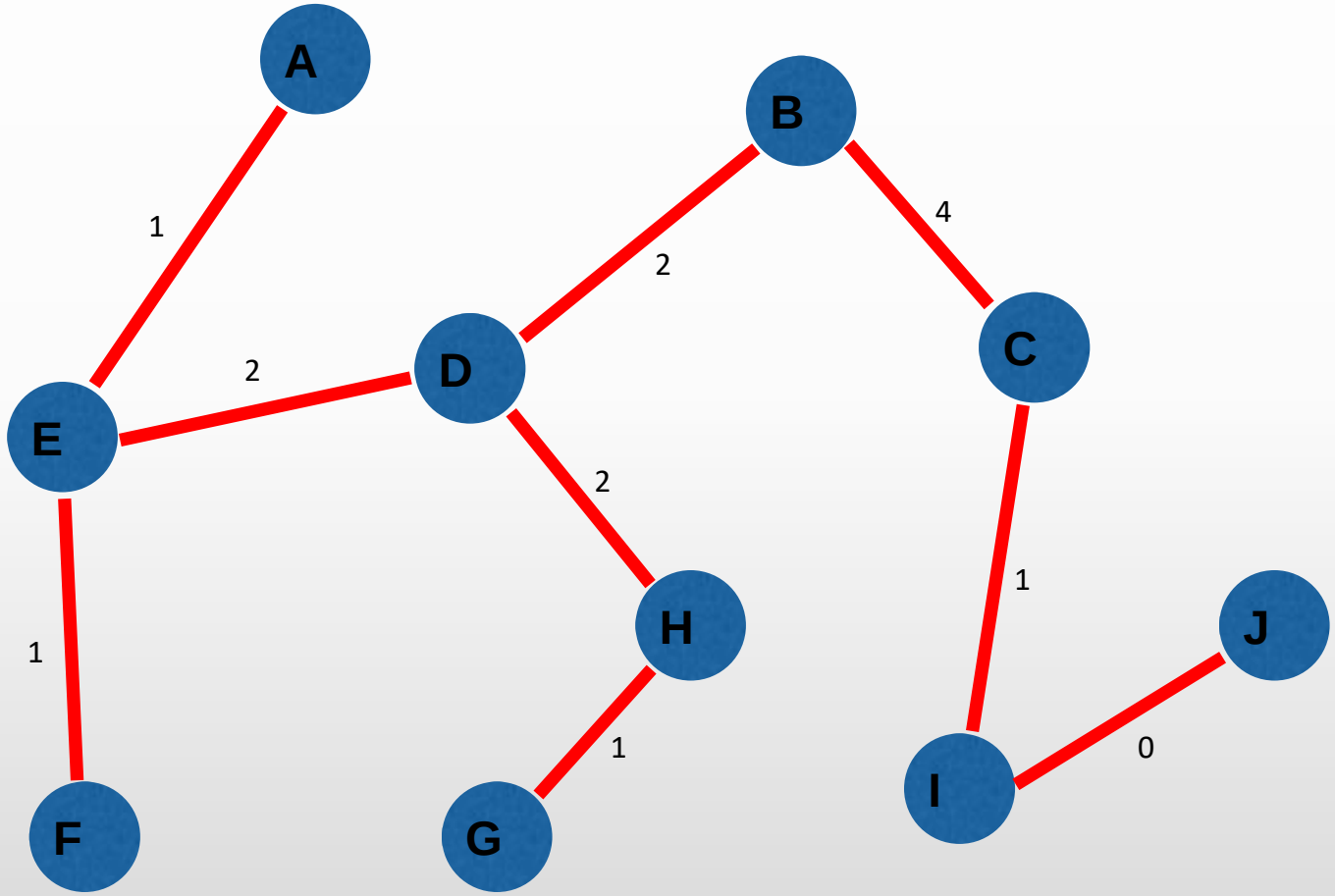
Kruskal Minimum Kapsayan Ağaç

- Çizgedeki tüm düğümleri minimum maliyetle kapsayan ağaçtır.
- Toplam kenar maliyeti en düşük olan kenarlar alt kümesidir.

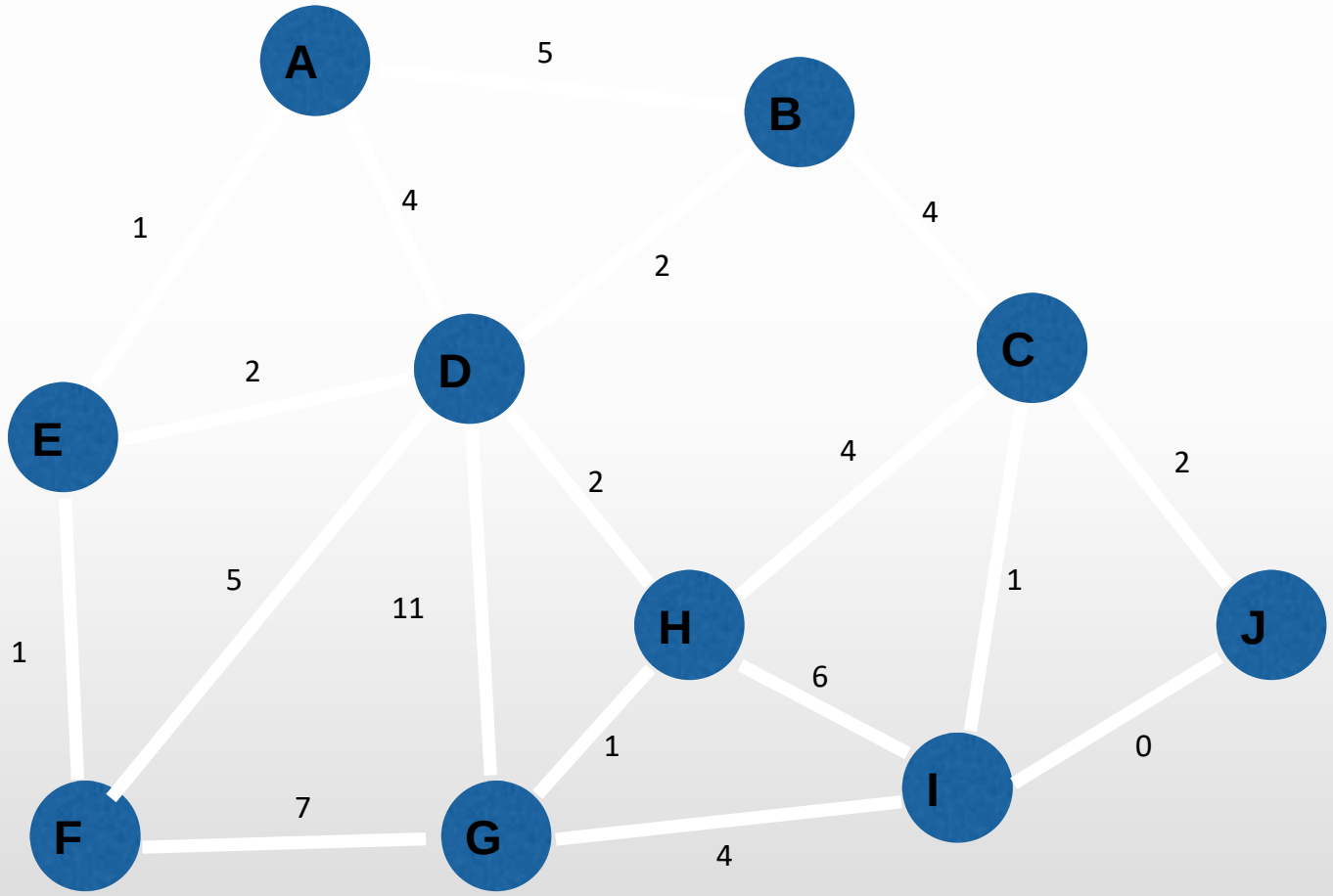
Kruskal Minimum Kapsayan Ağaç



Kruskal Minimum Kapsayan Ağaç

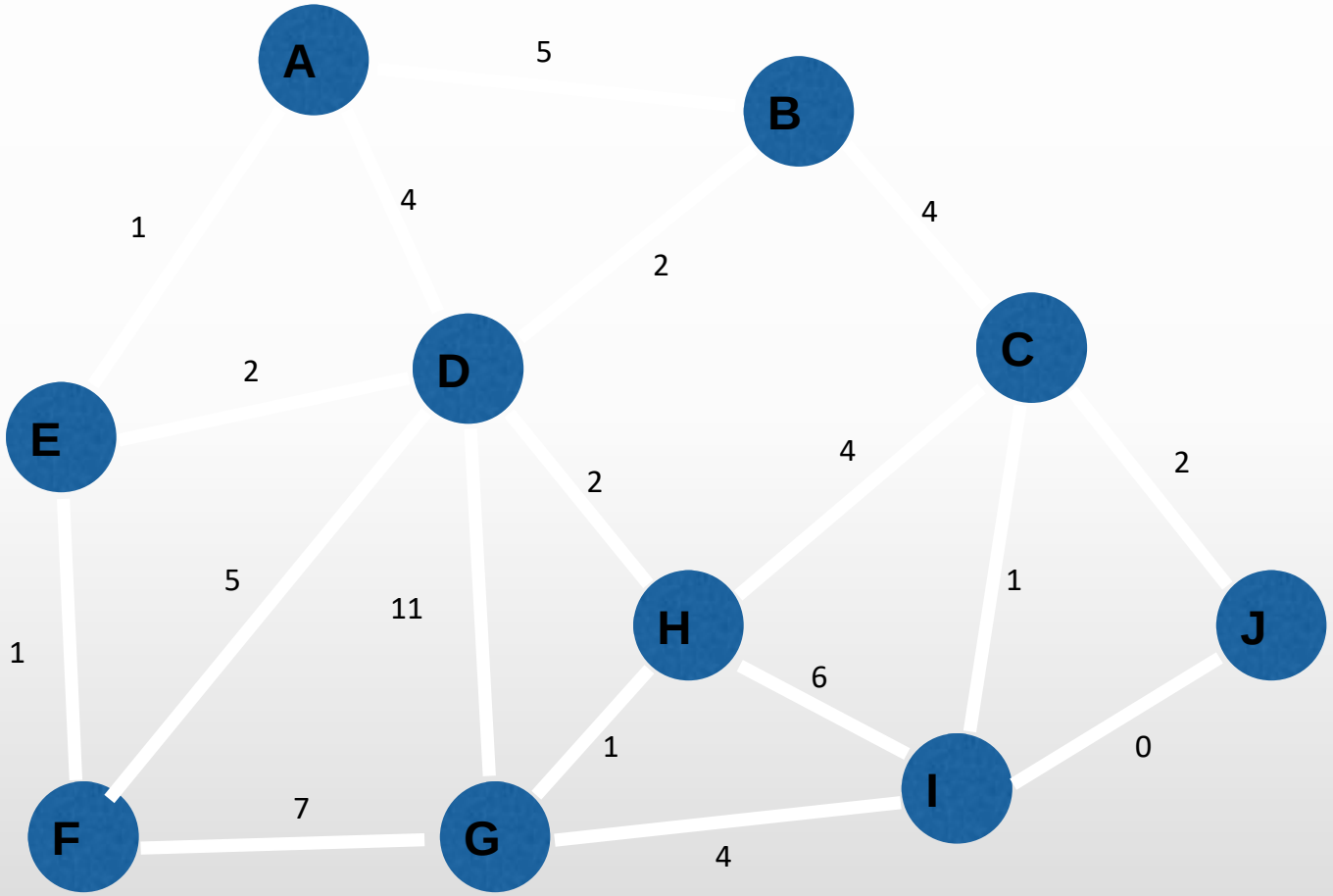


Kruskal Algoritması Uygulama



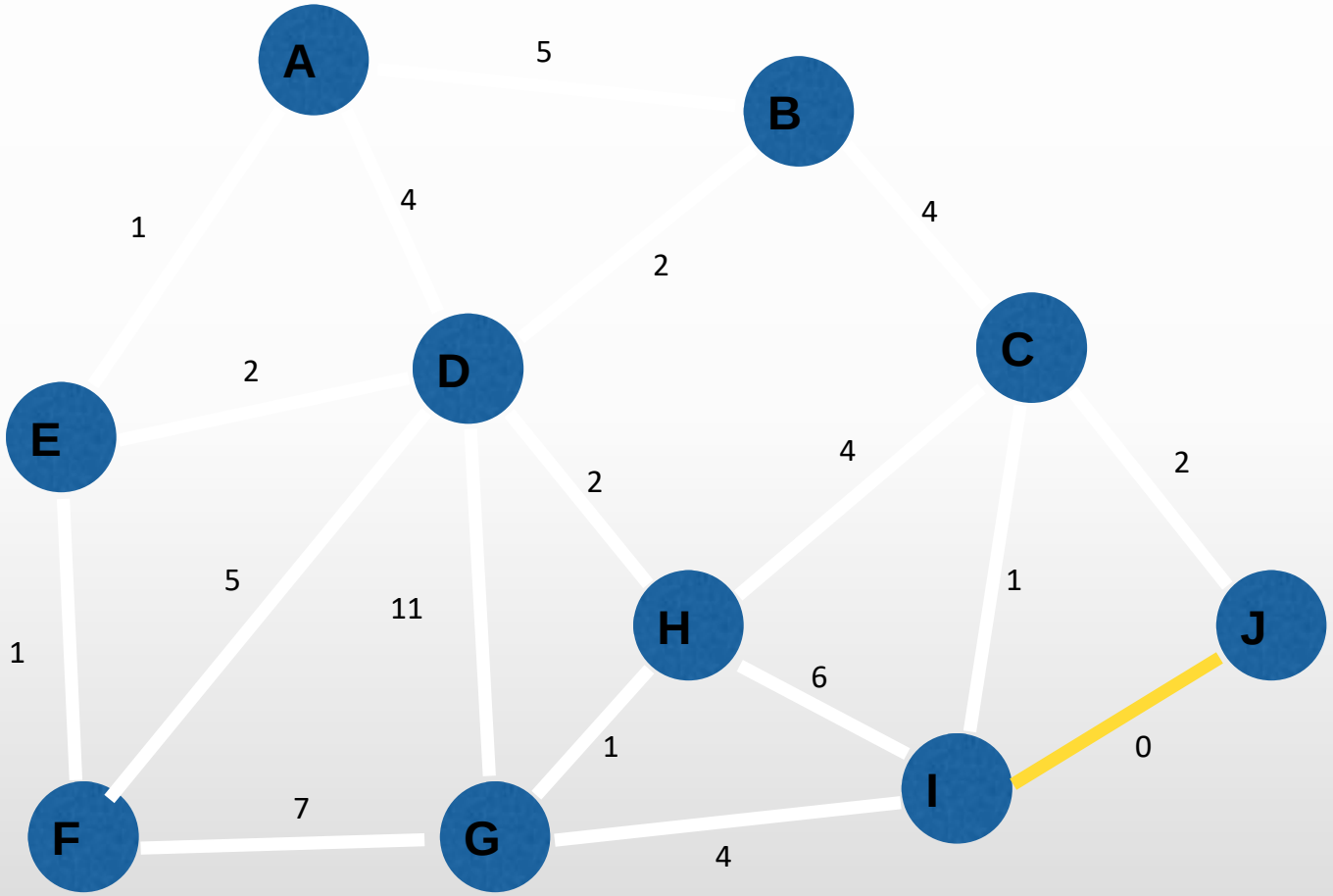
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



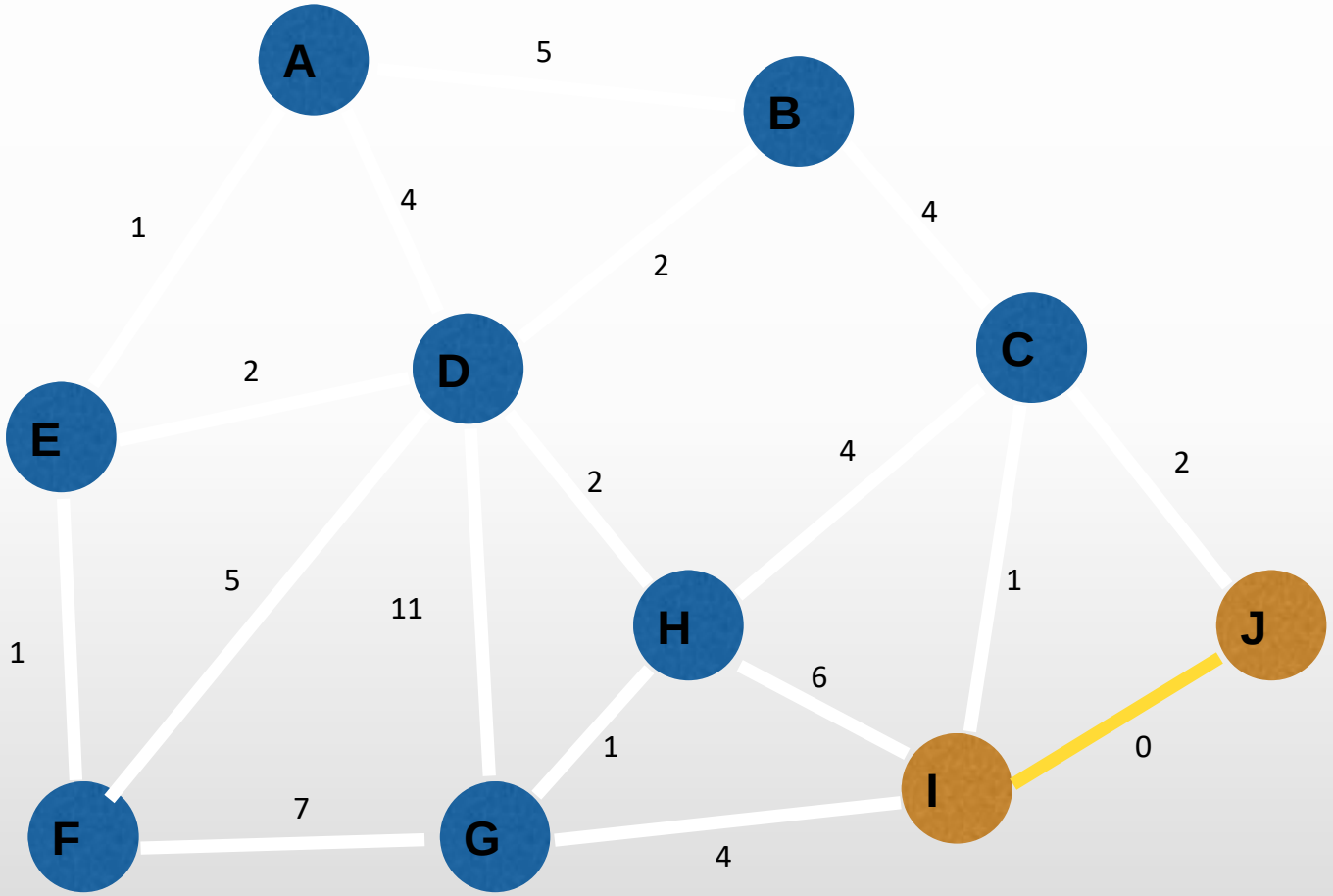
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



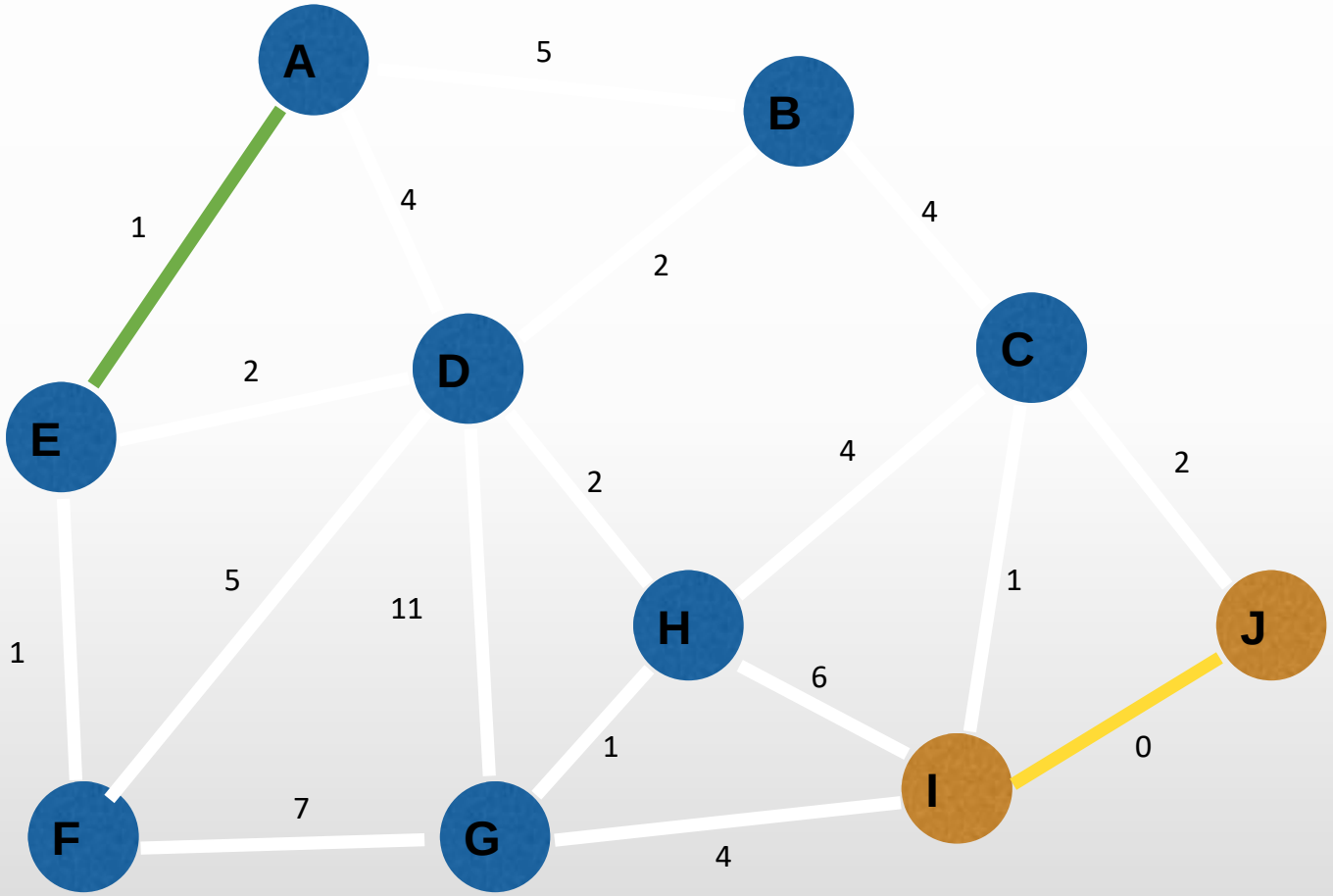
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



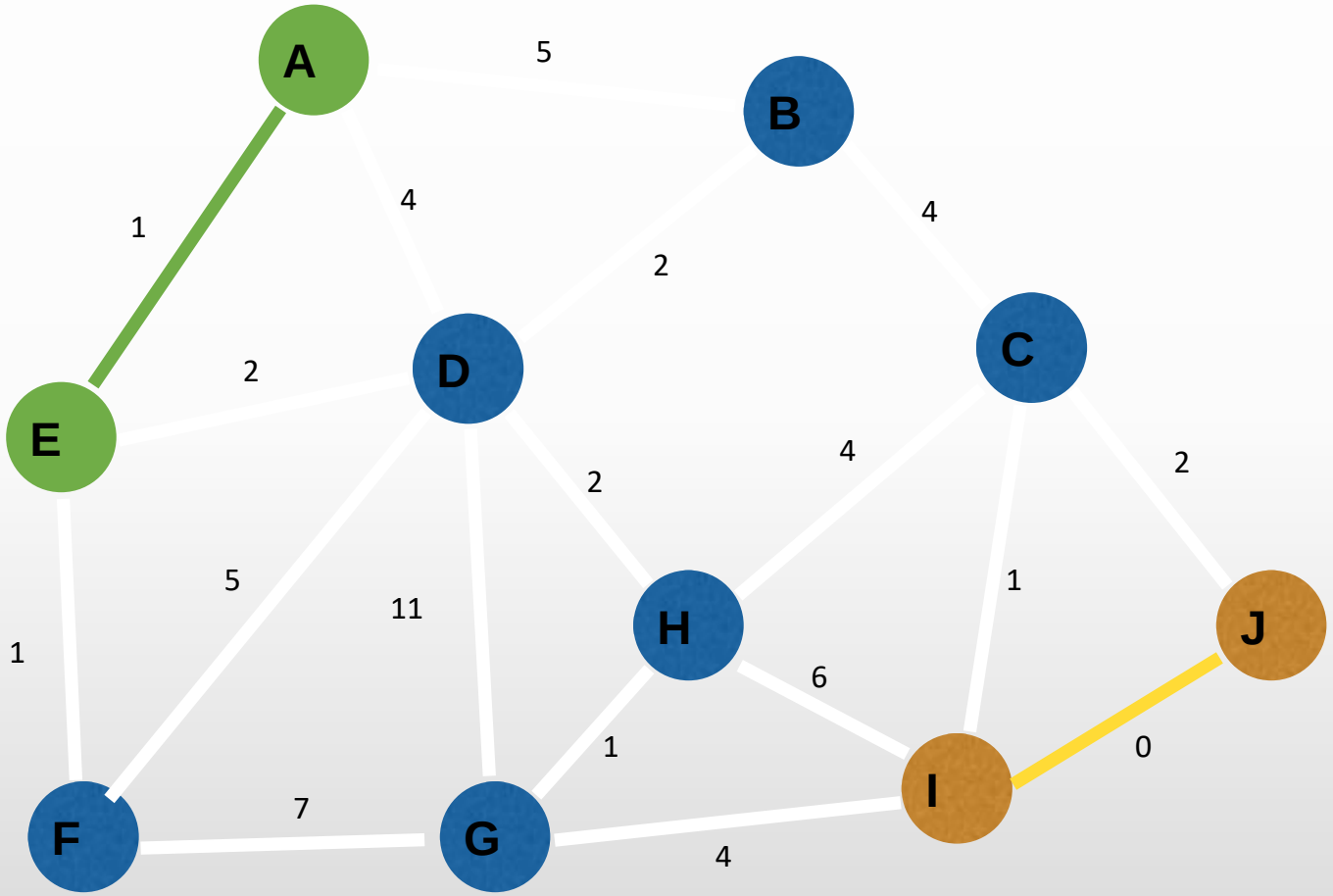
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



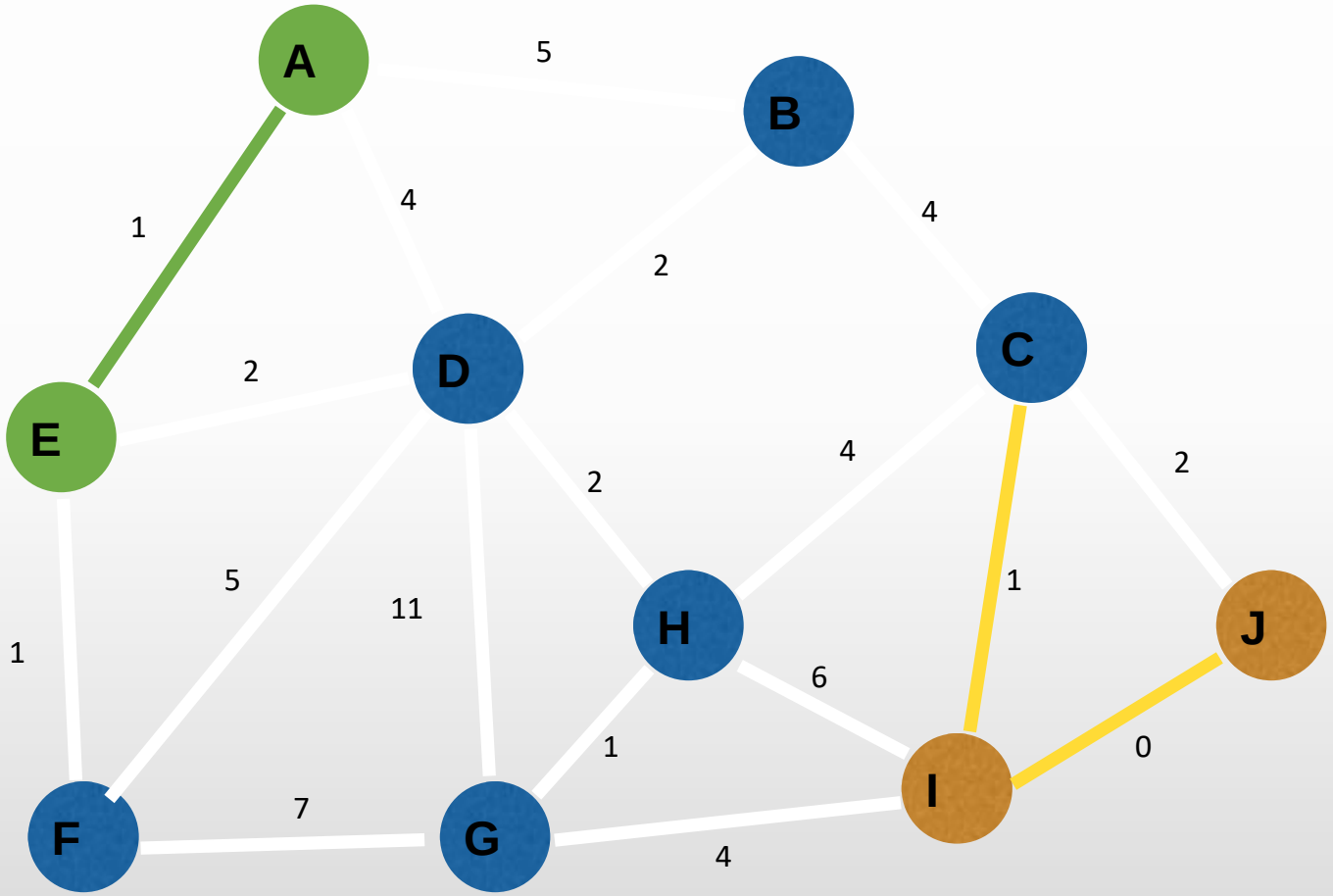
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



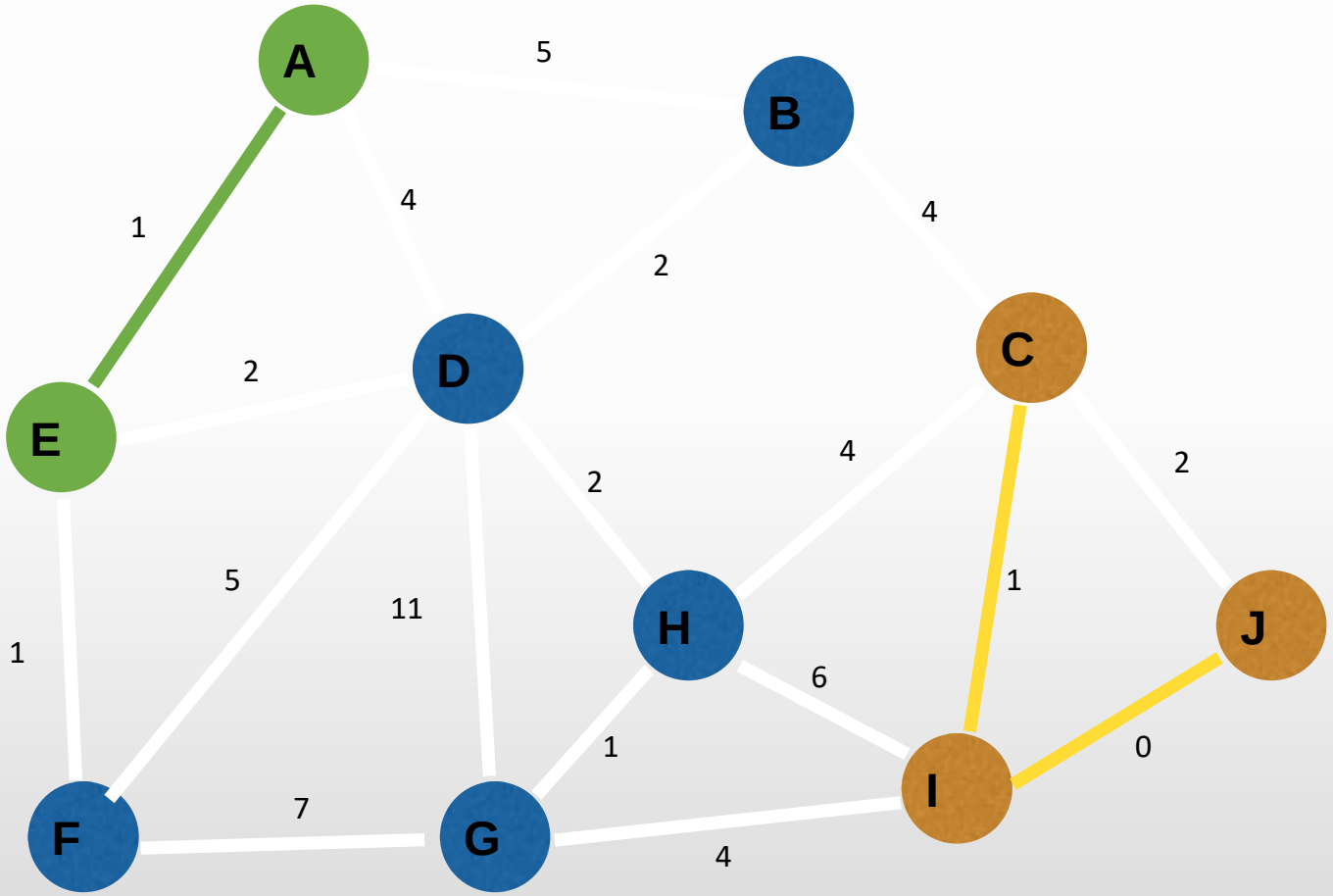
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



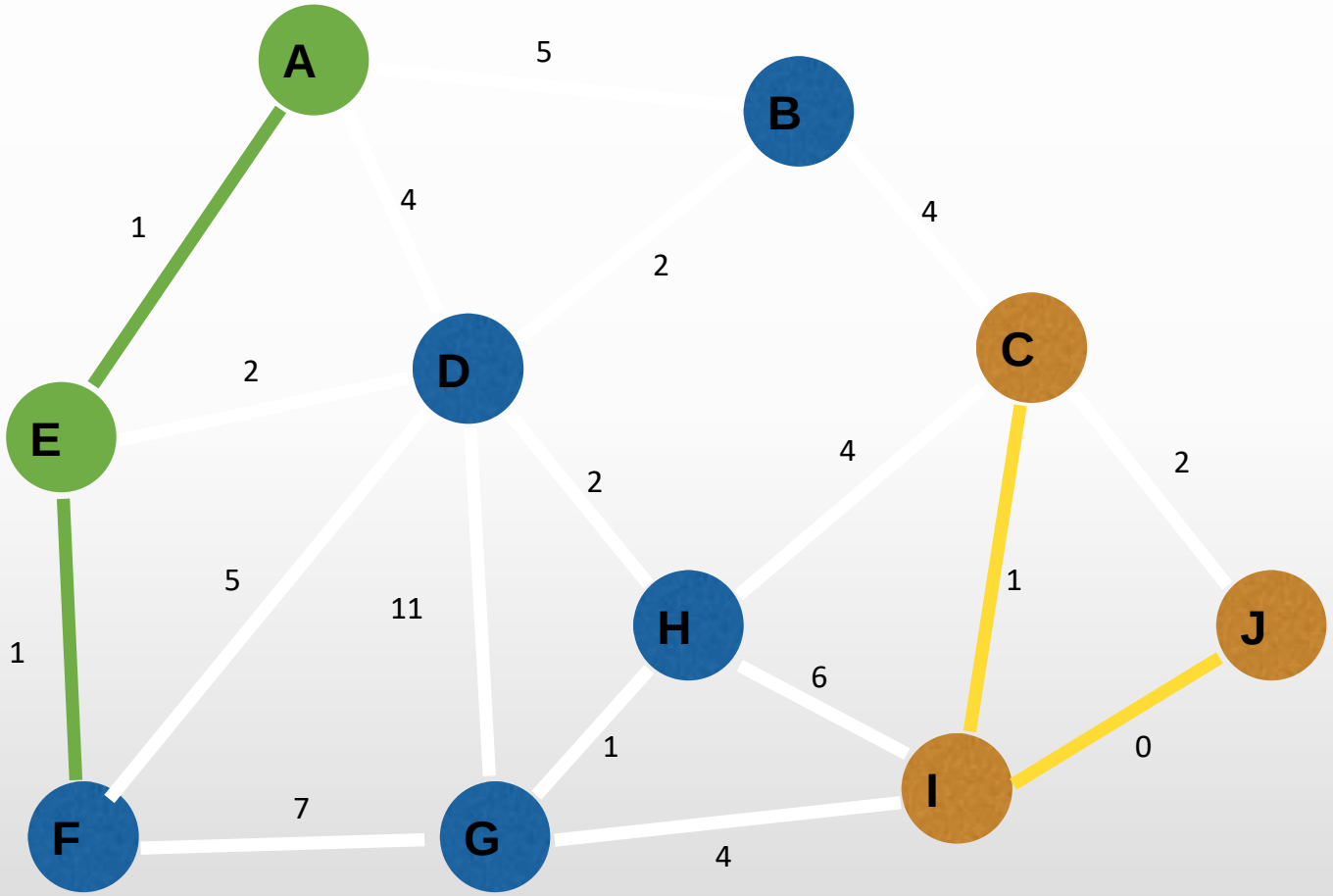
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



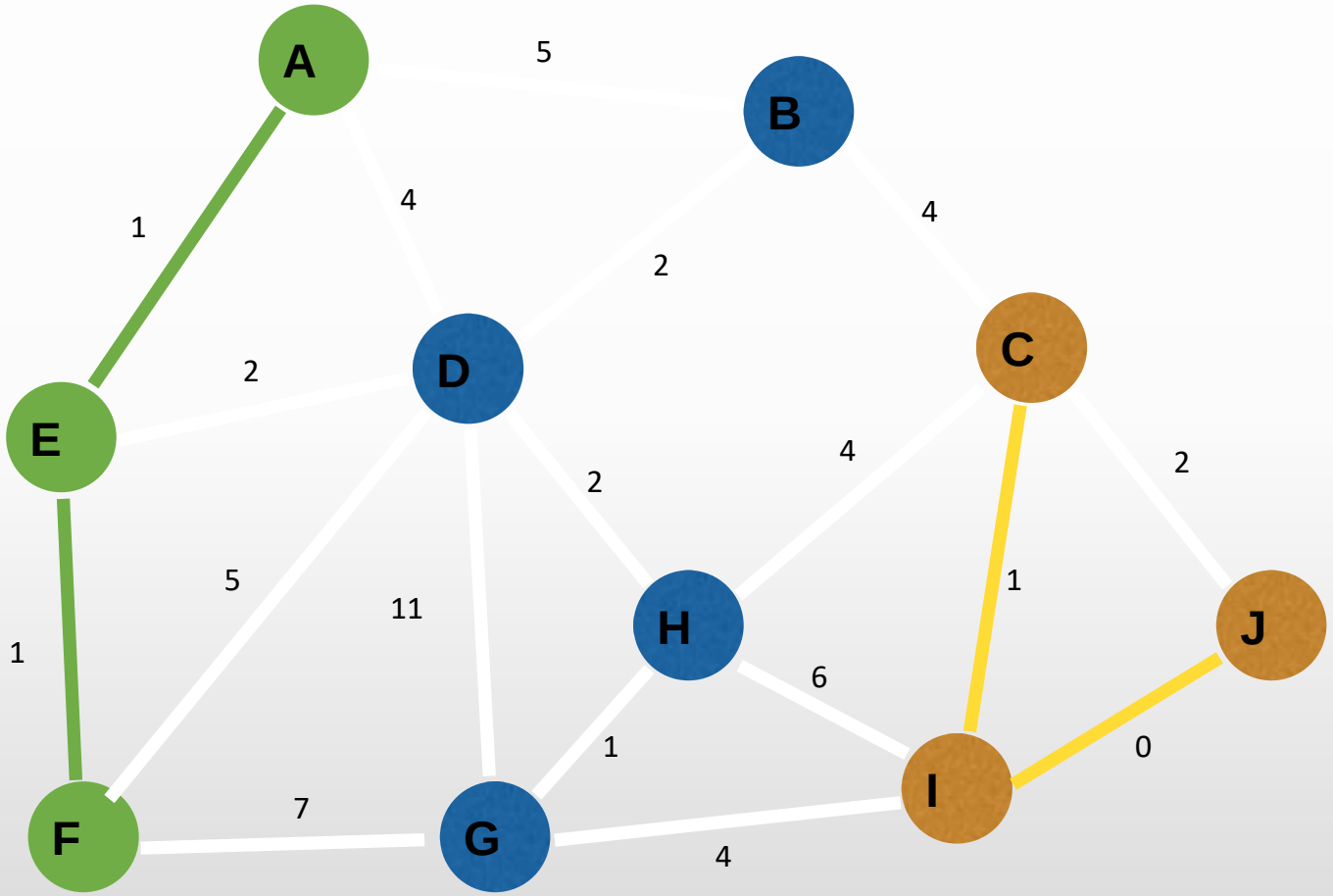
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



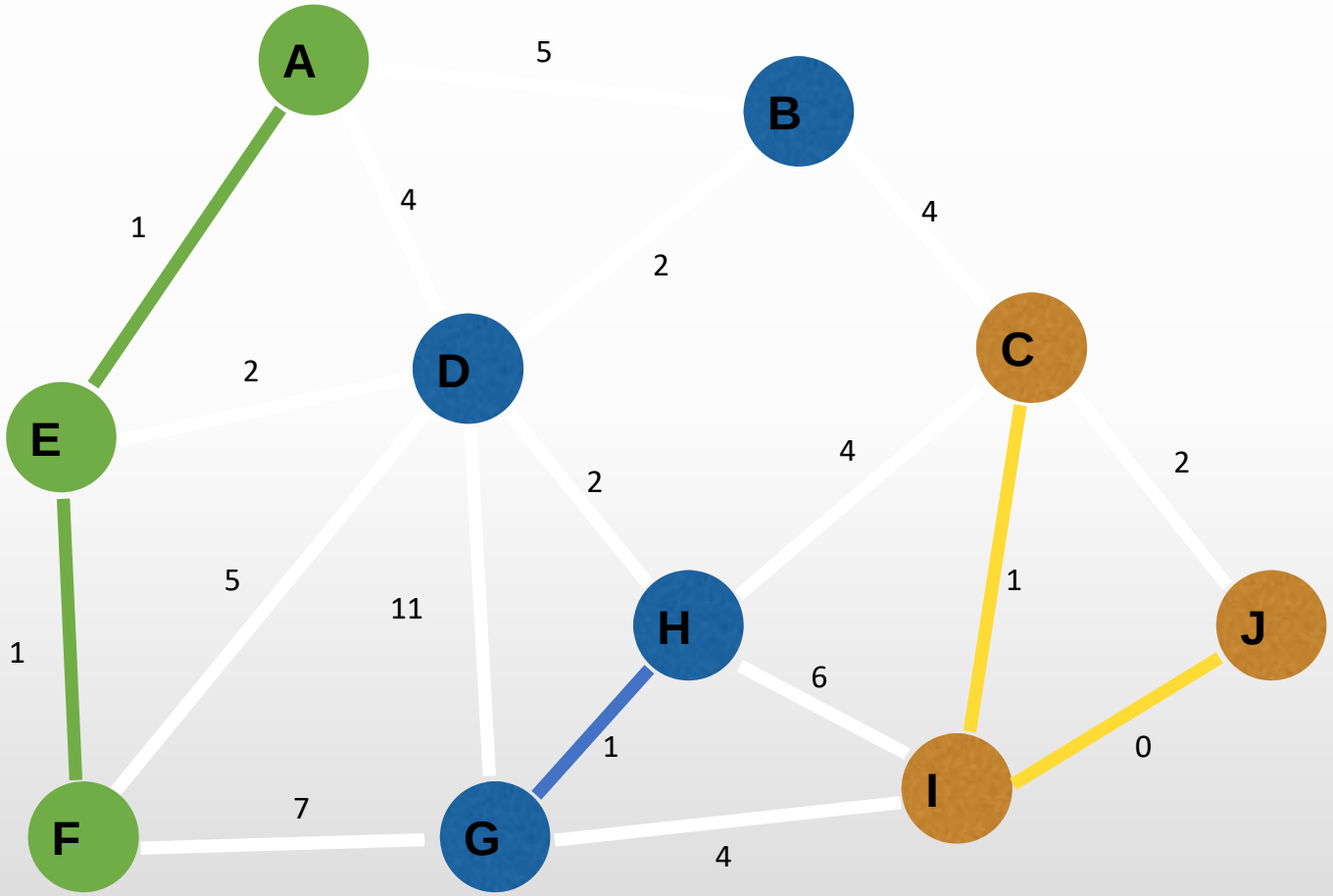
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



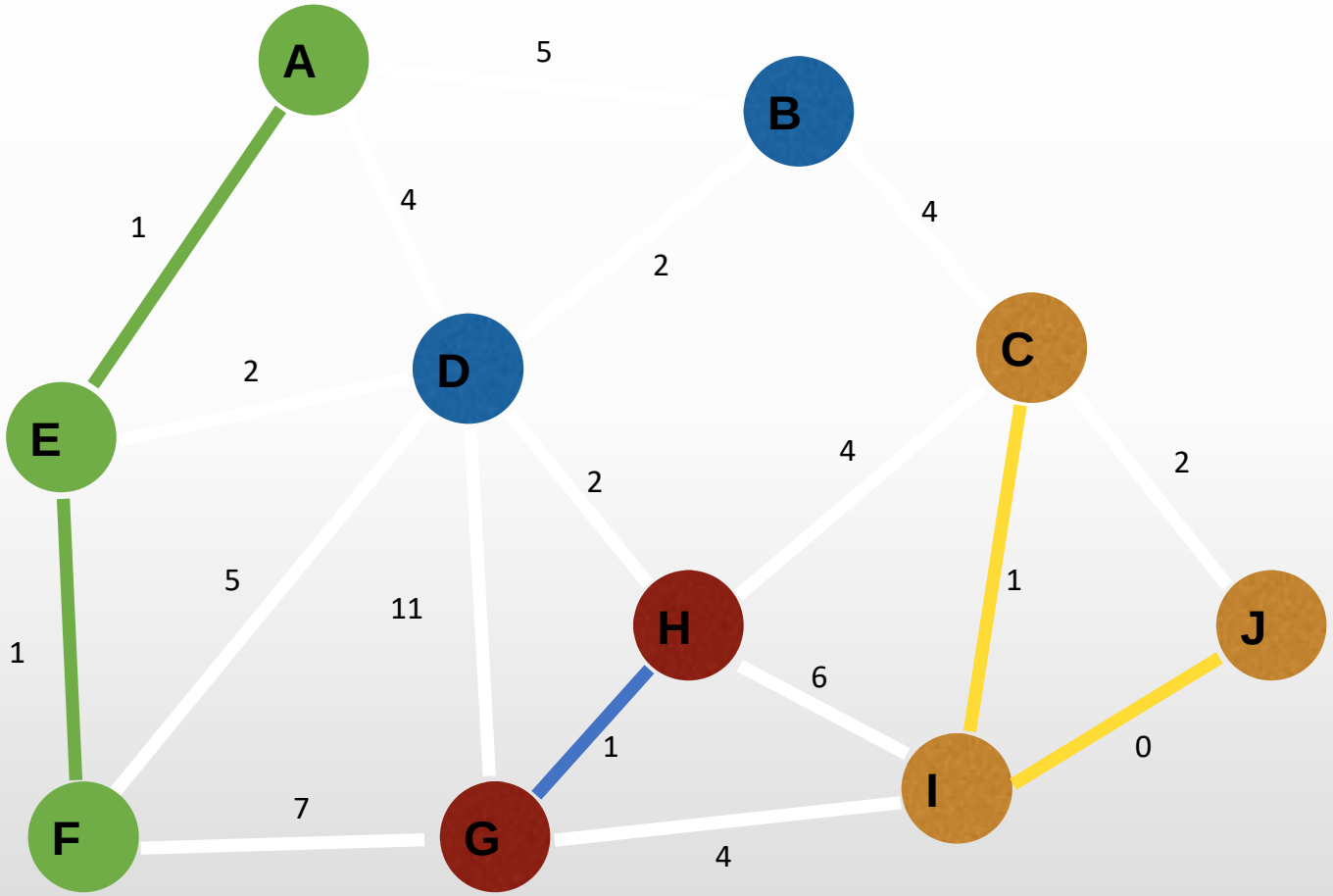
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



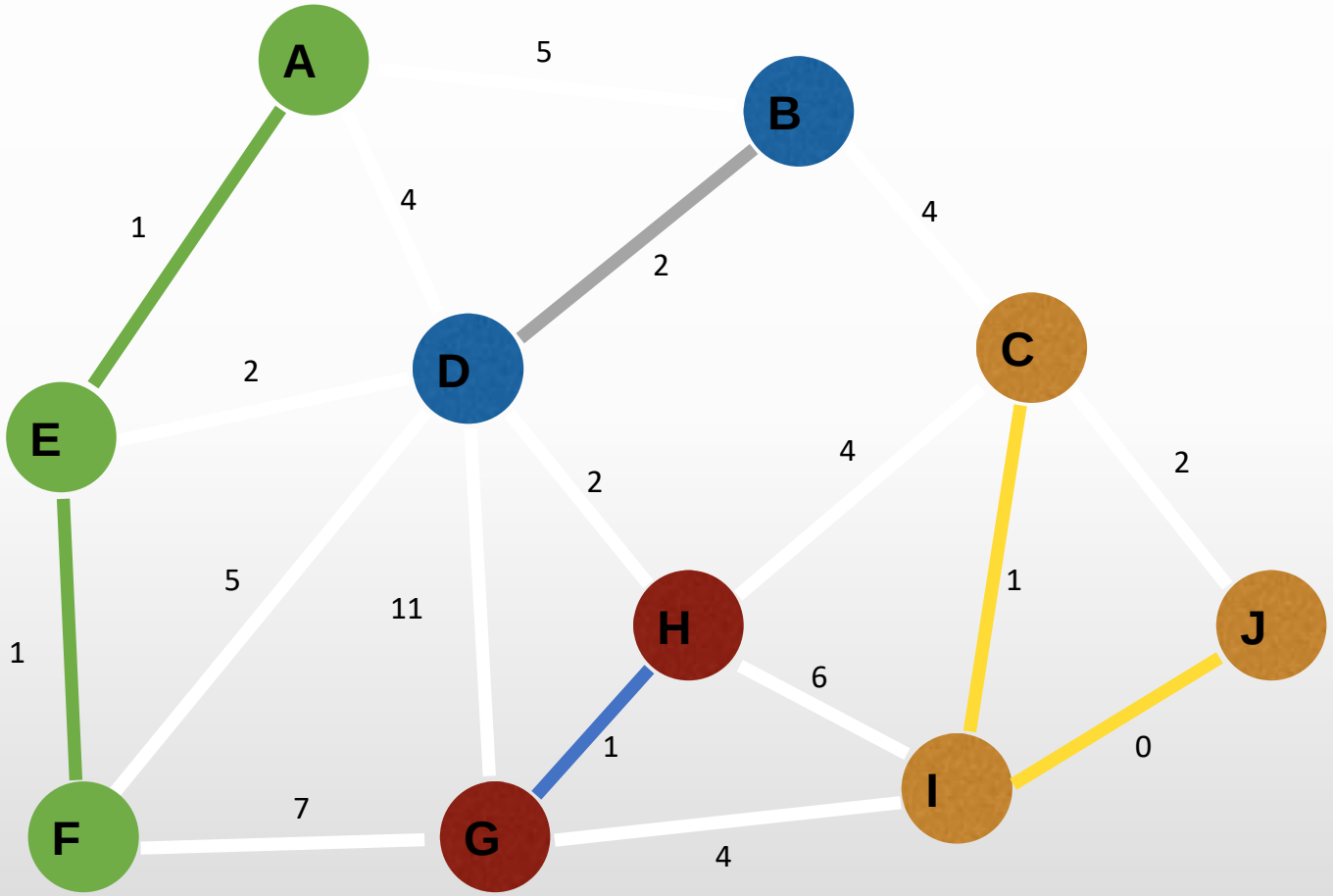
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



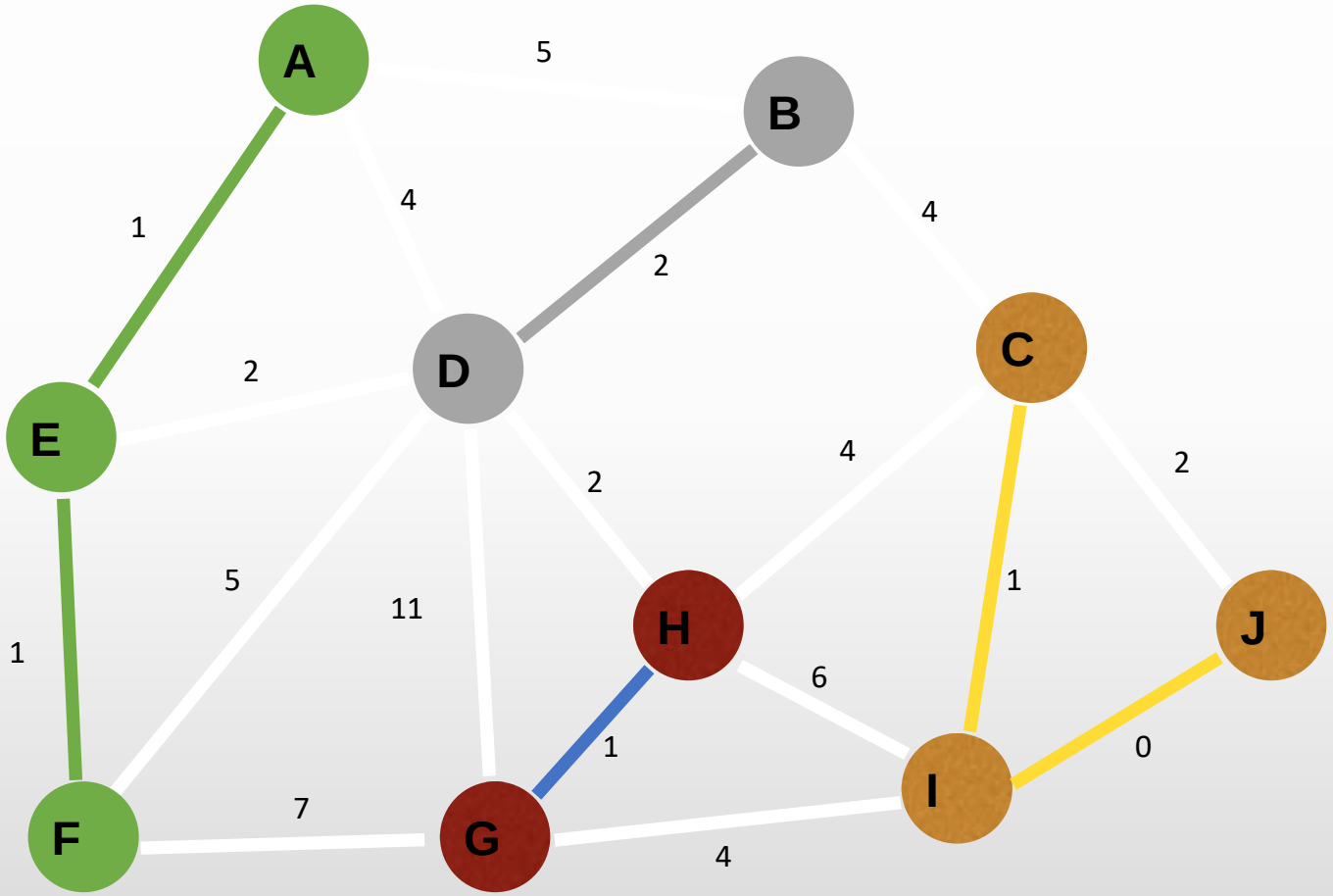
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



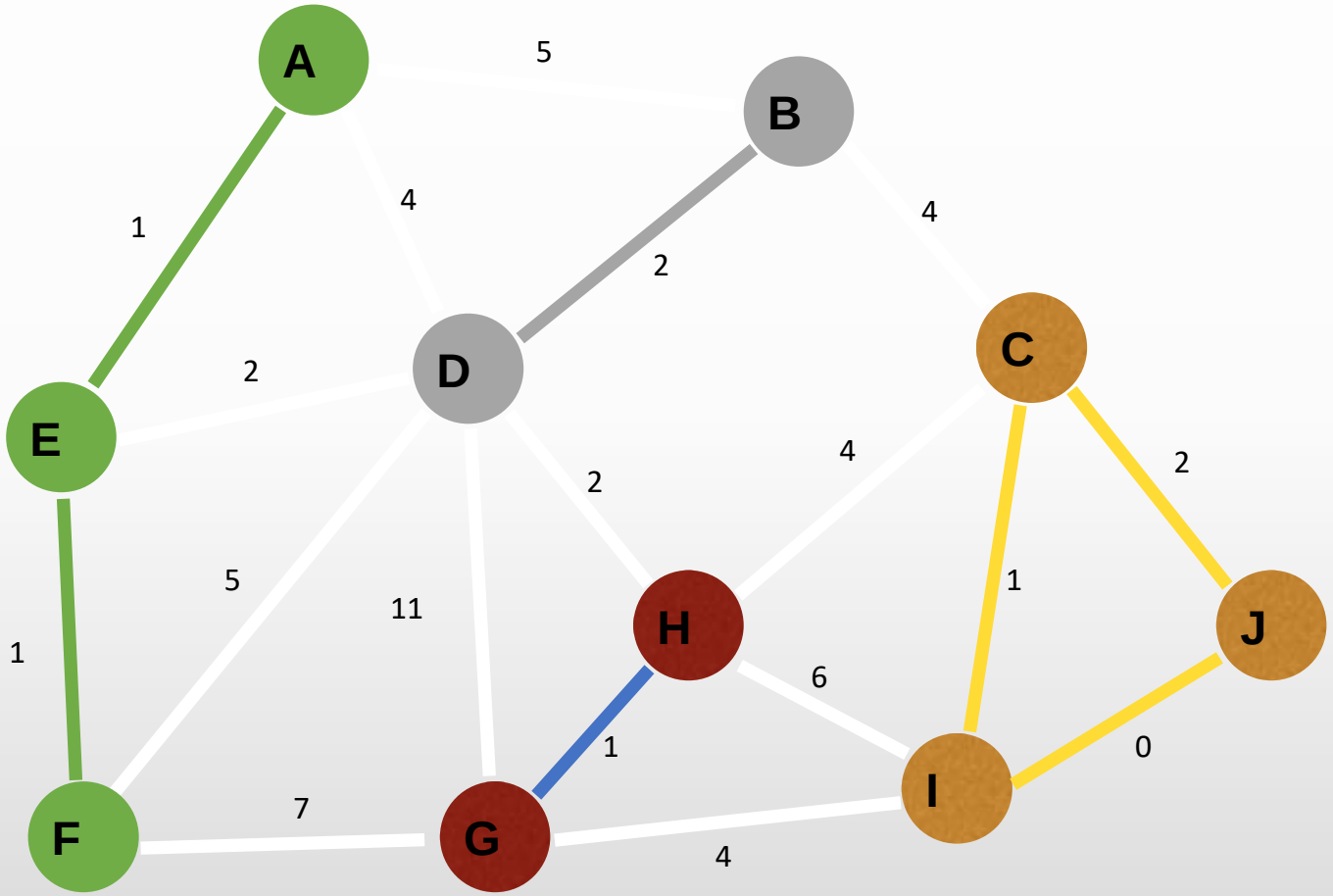
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



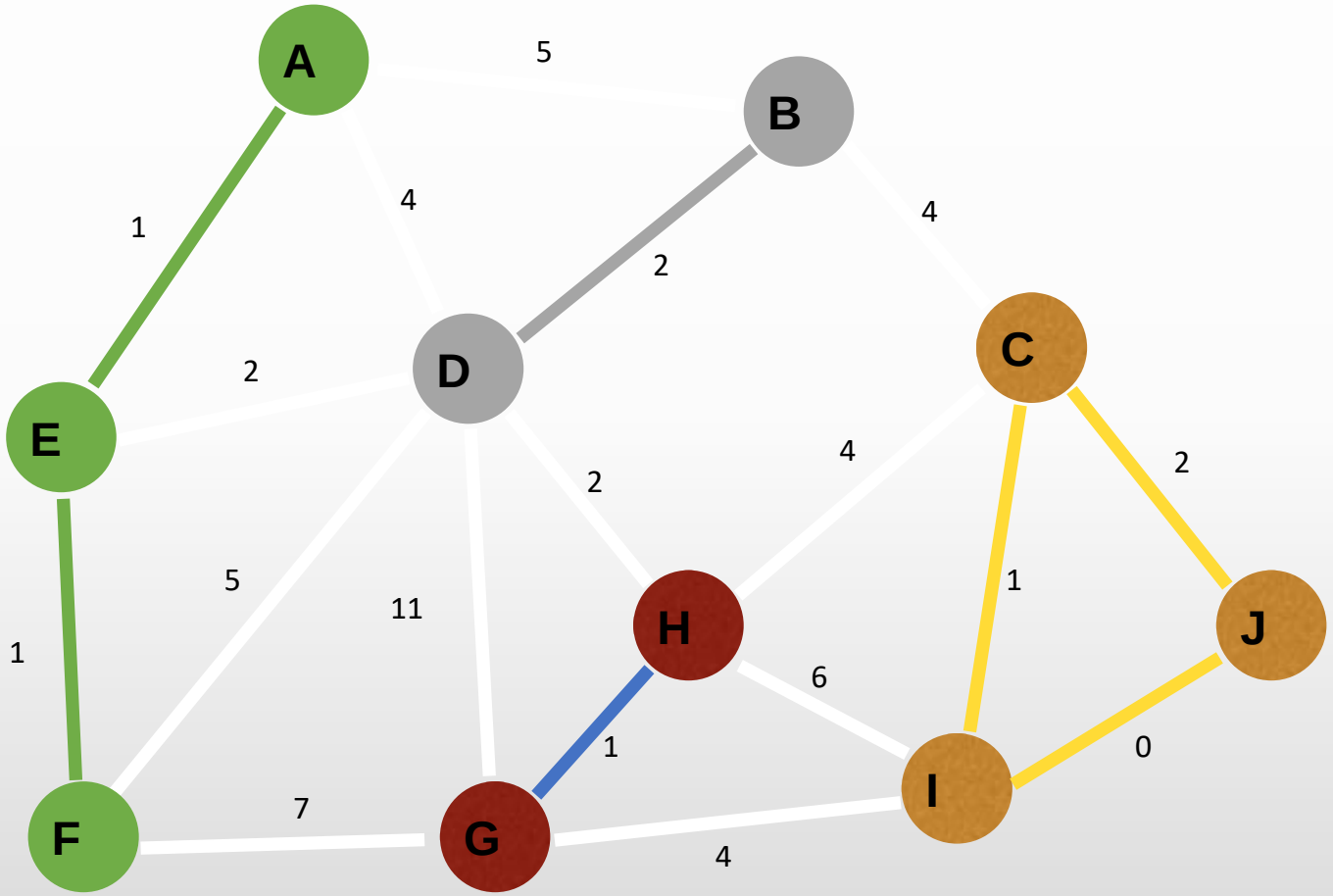
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



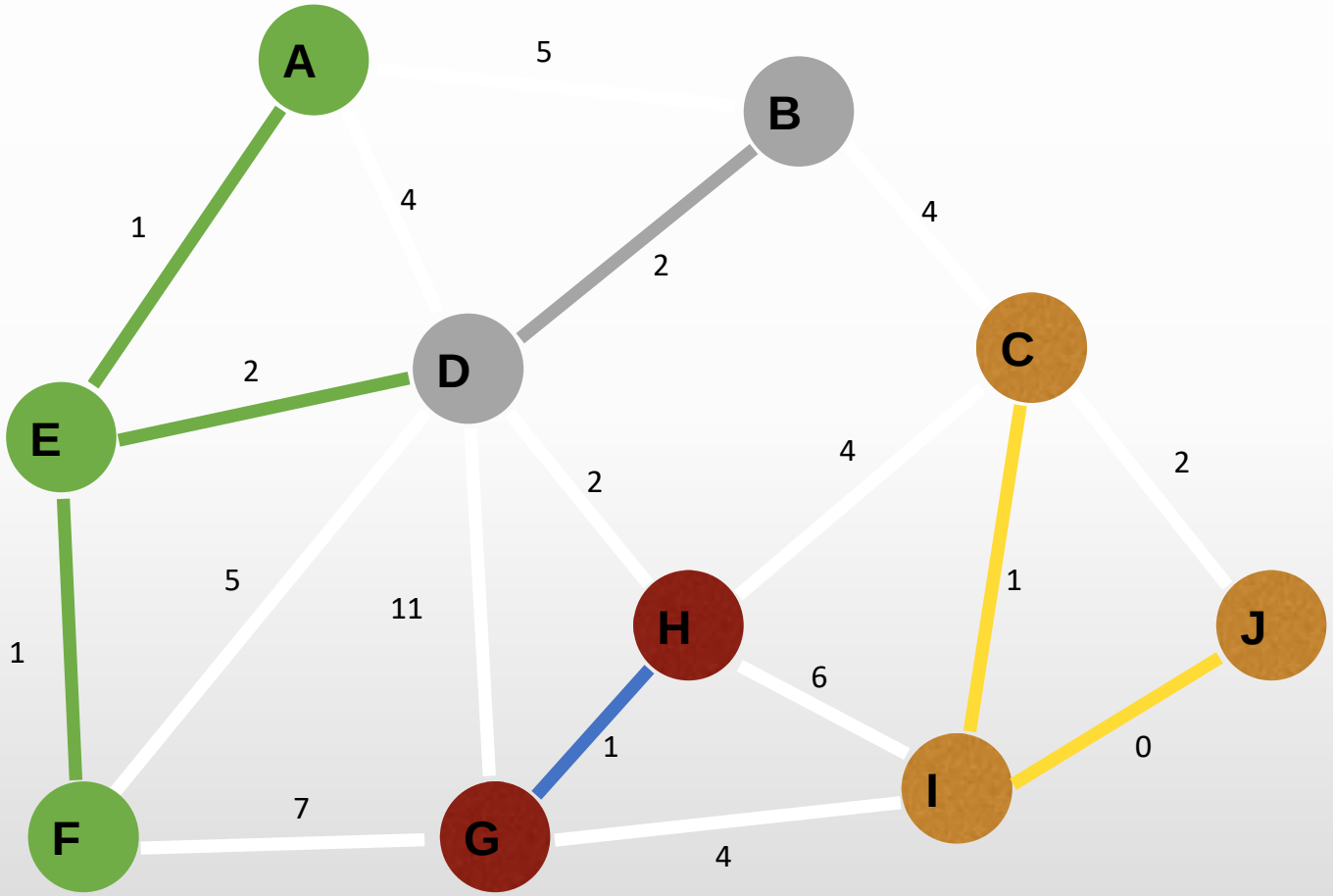
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



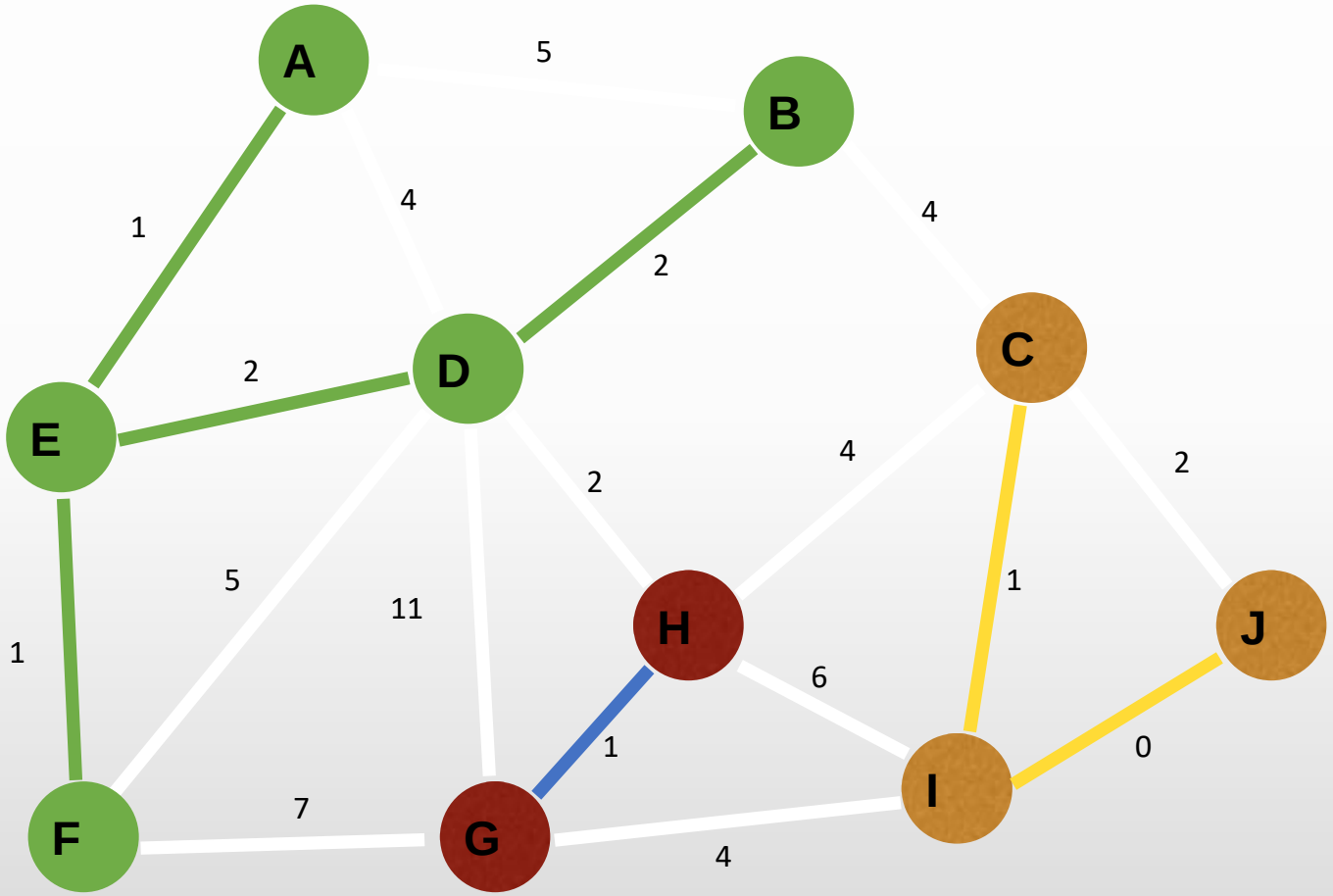
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



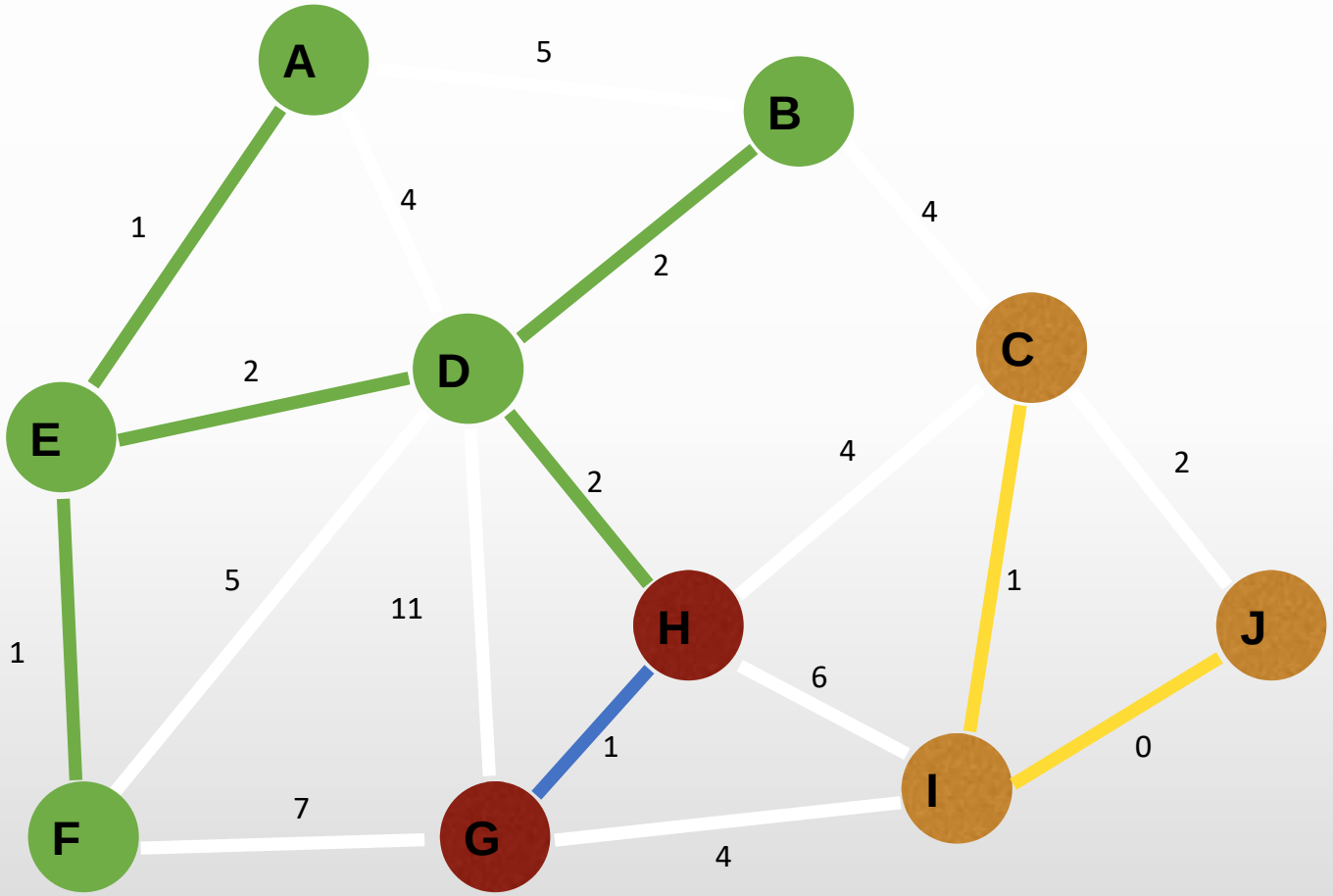
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



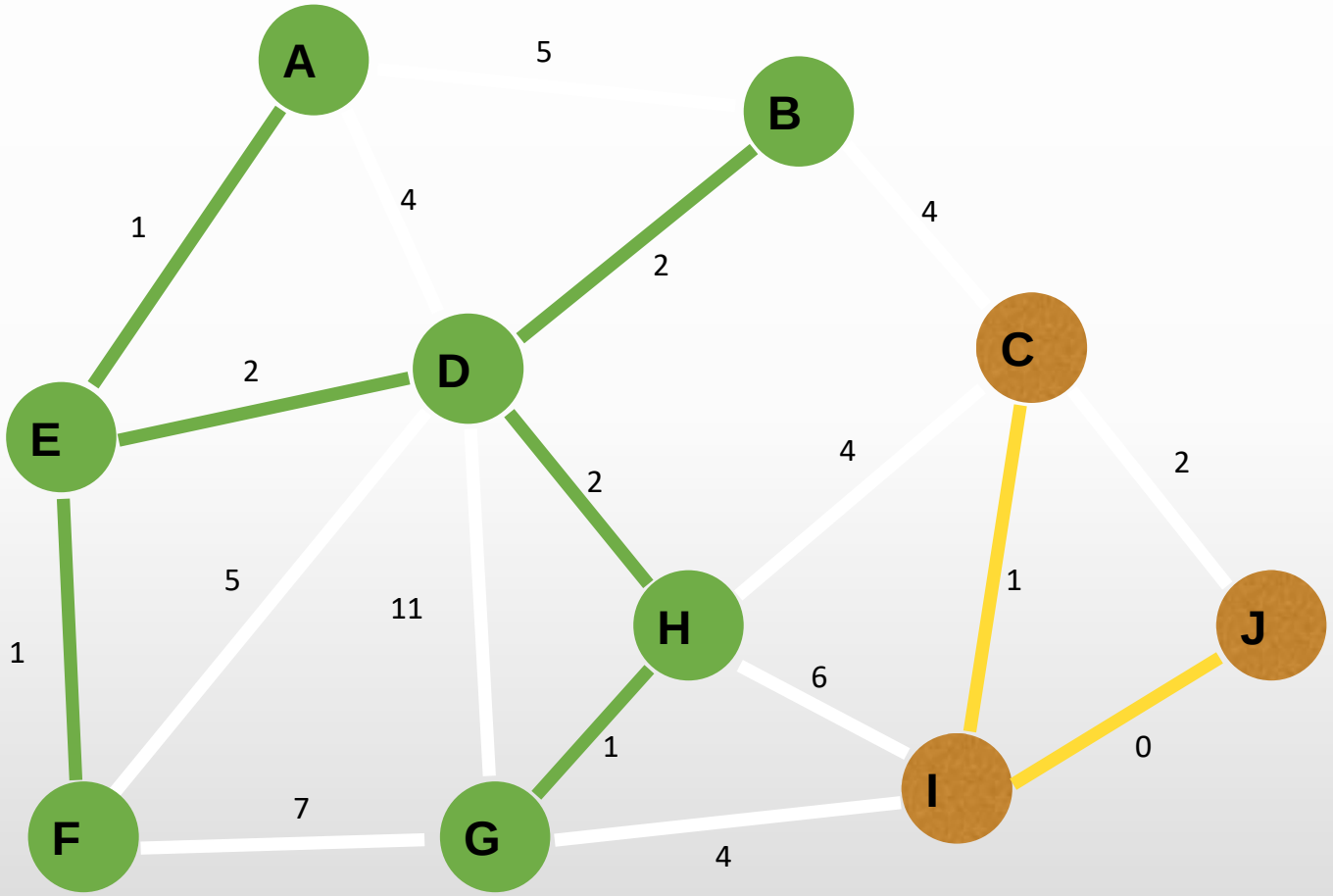
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



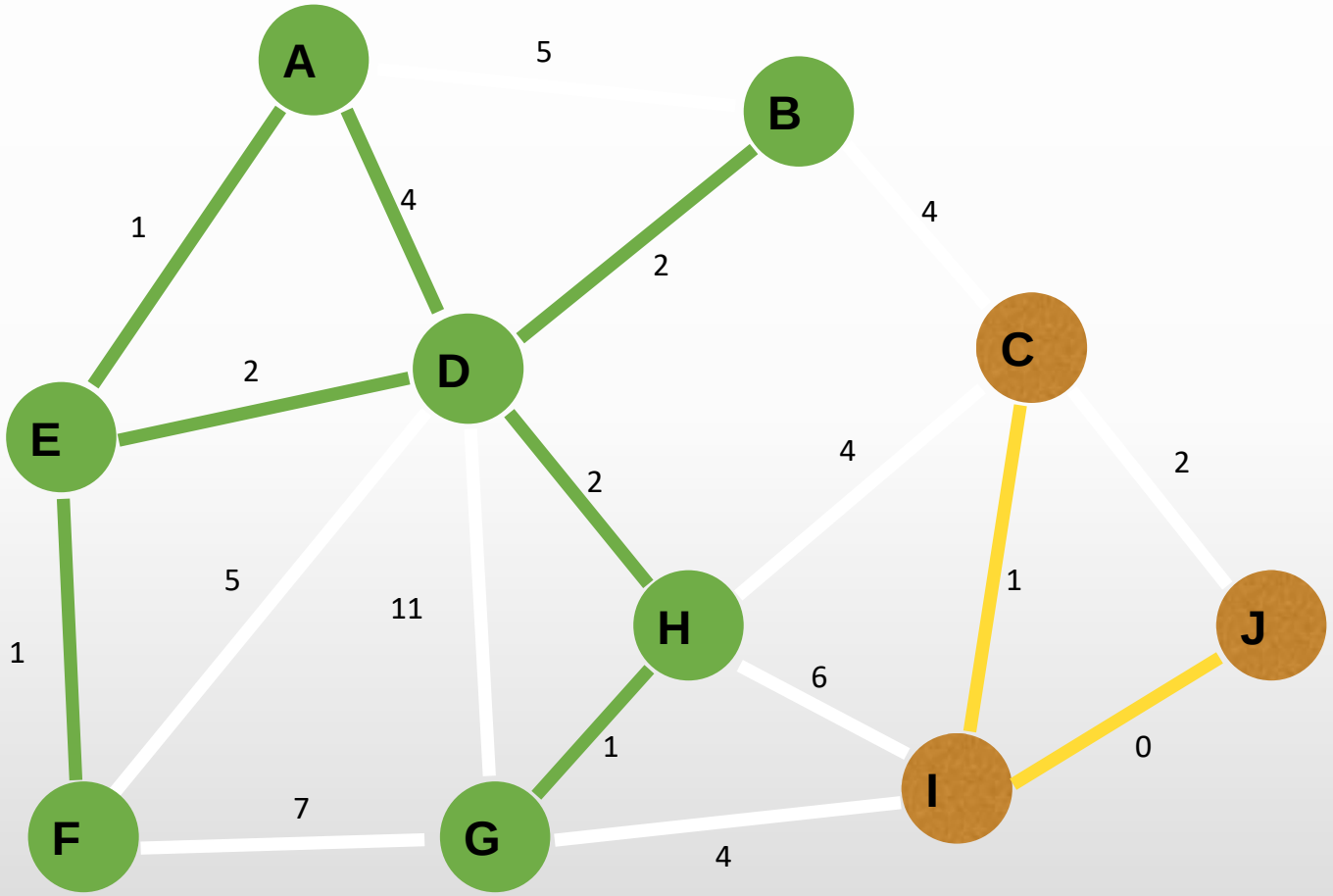
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



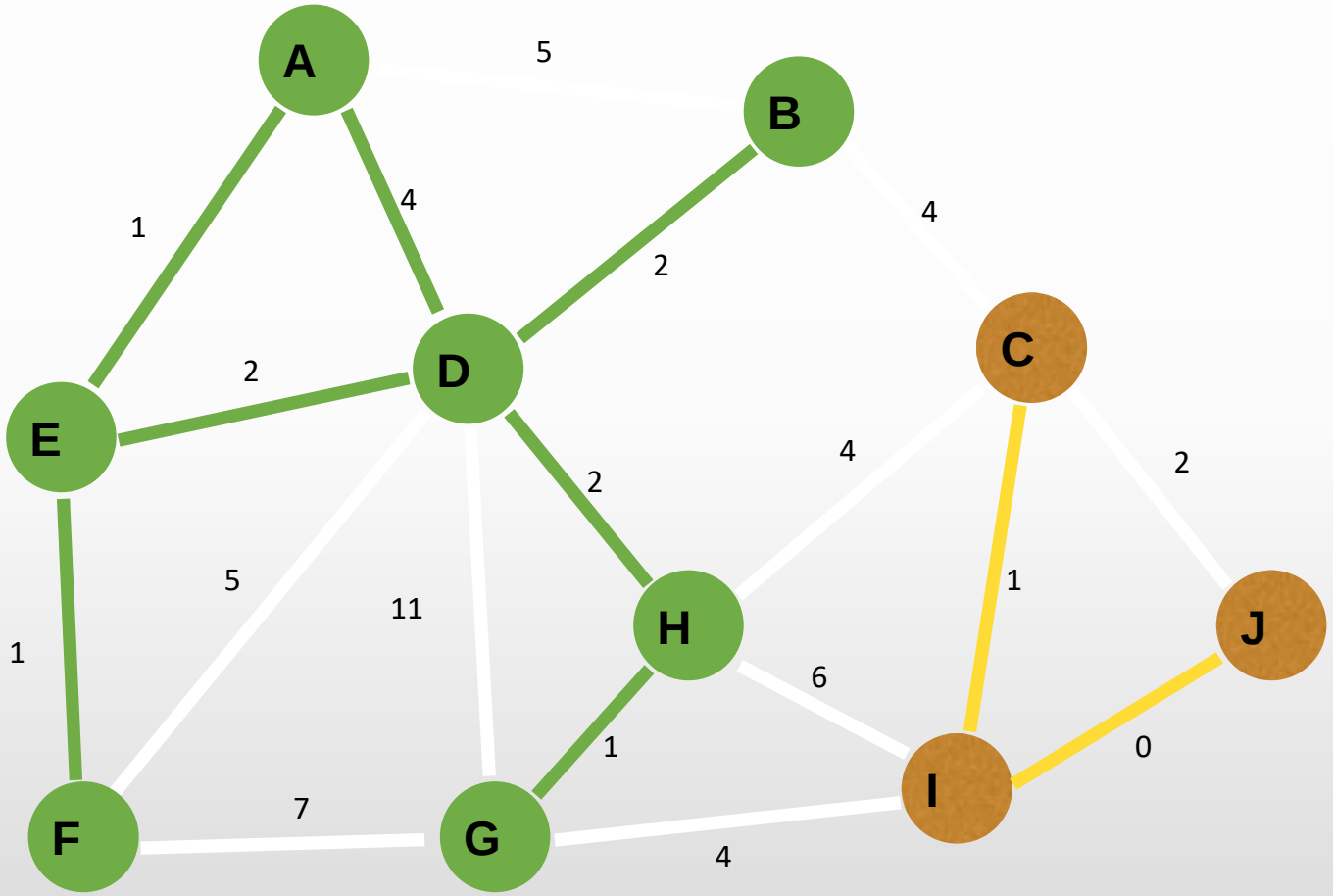
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



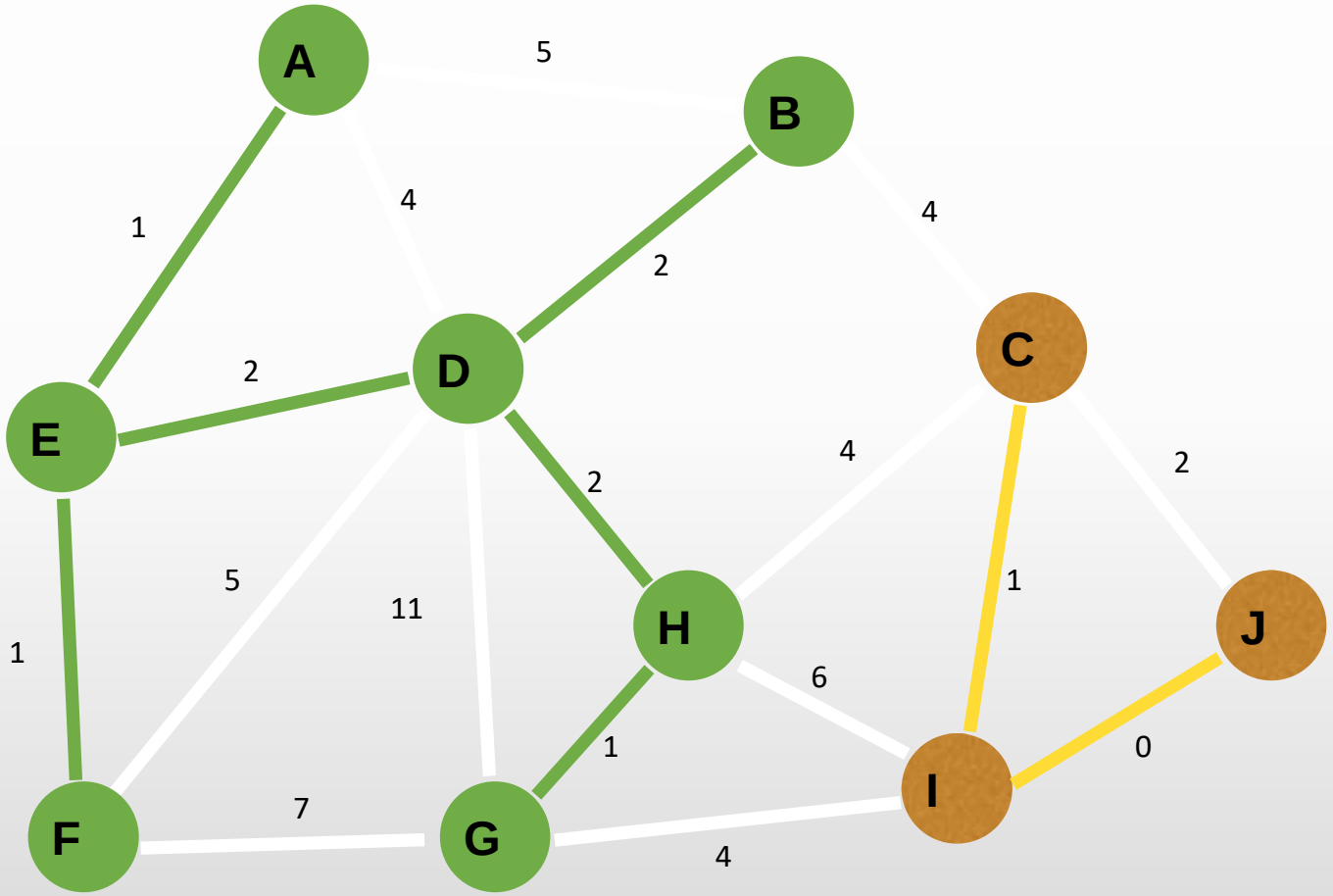
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



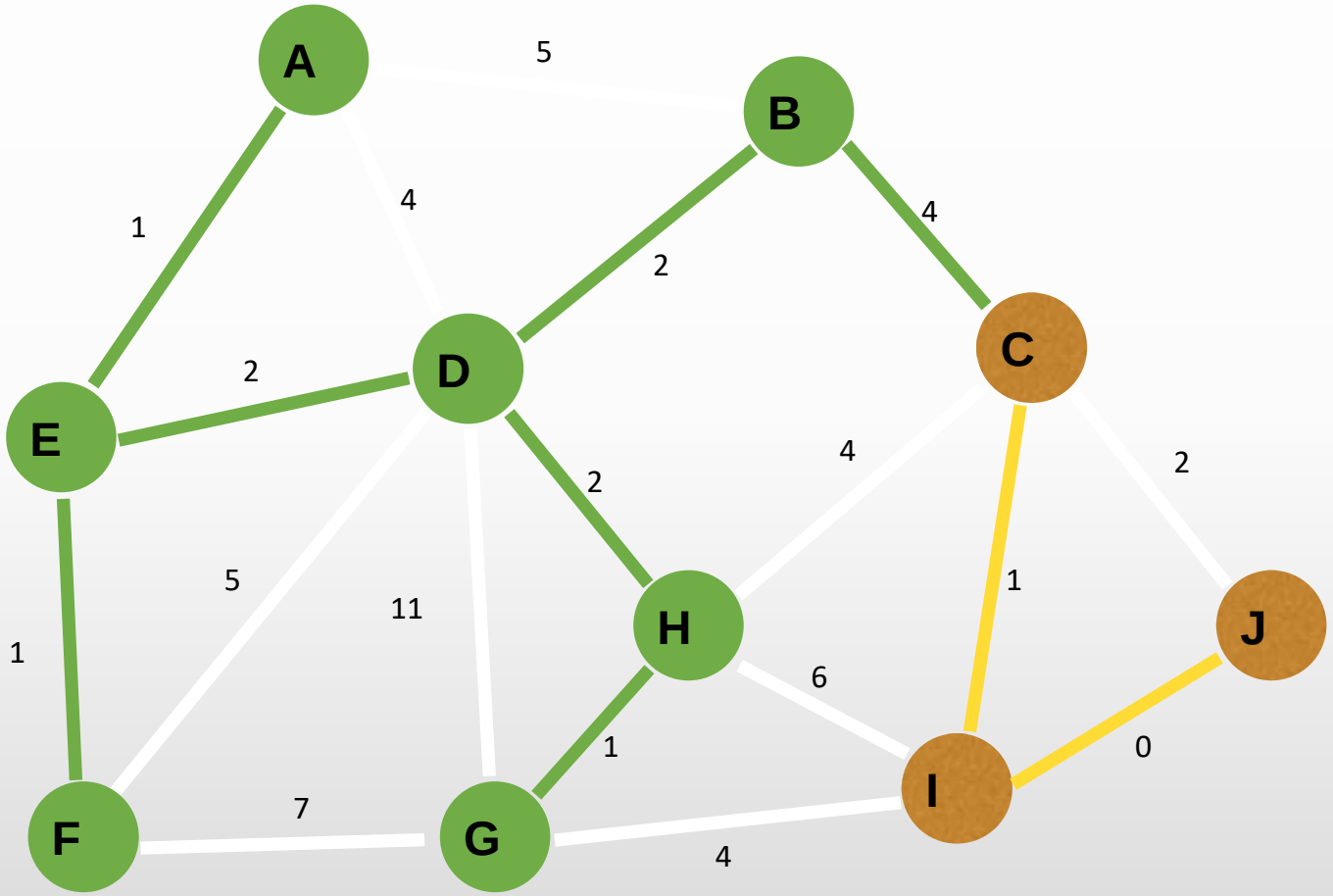
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



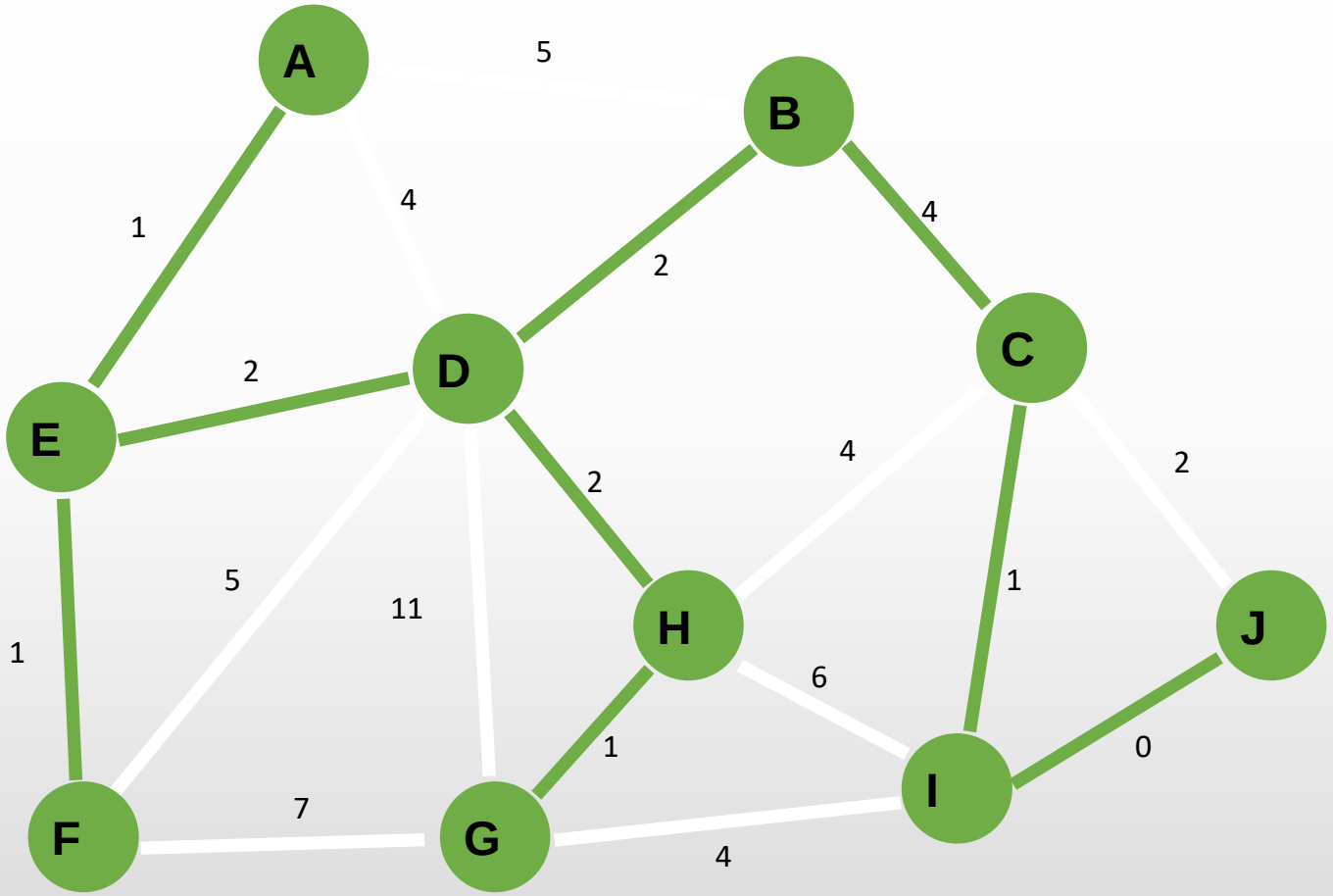
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



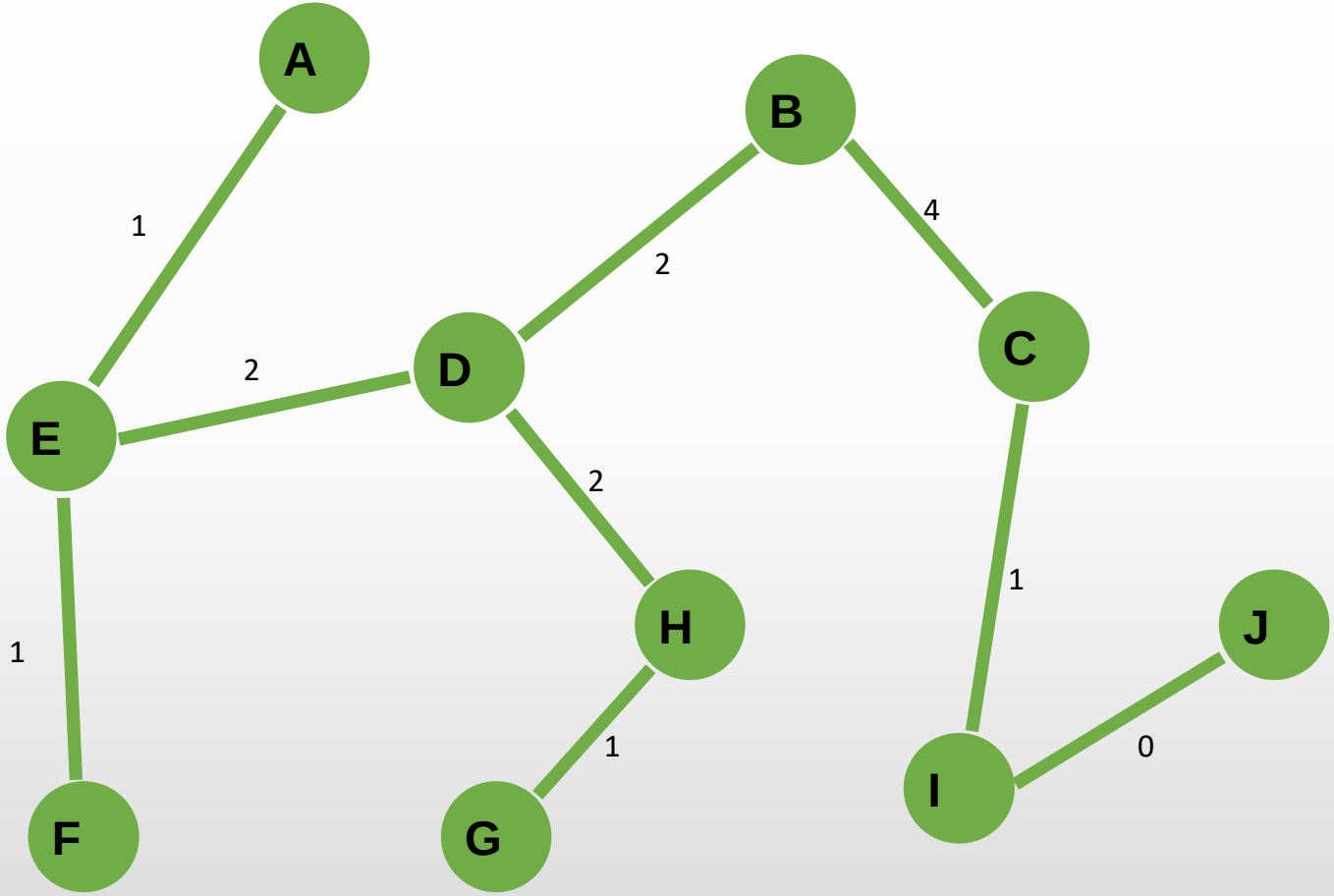
Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



Kruskal Algoritması Uygulama

```
I -> J = 0
A -> E = 1
C -> I = 1
E -> F = 1
G -> H = 1
B -> D = 2
C -> J = 2
D -> E = 2
D -> H = 2
A -> D = 4
B -> C = 4
C -> H = 4
G -> I = 4
A -> B = 5
D -> F = 5
H -> I = 6
F -> G = 7
D -> G = 11
```



Küme İşleçleri

- **Birleştirme (Union):** İki kümenin elemanlarından bir küme oluşturur.
- **Kesişim (Intersection):** İki kümenin ortak elemanlarından bir küme oluşturur.
- **Fark (Difference):** Bir kümenin diğer kümede olmayan elemanlarından bir küme oluşturur.
- **Alt küme (Subset):** Bir kümenin diğer bir kümenin alt kümesi olup olmadığını söyler.

Kümelerin Gösterimi

- **A** = {1, 2, 3, 4, 5}
- **B** = {3, 4, 5, 6, 7}
- **Birleştirme**: A ∪ B = {1, 2, 3, 4, 5, 6, 7}
- **Kesişim**: A ∩ B = {3, 4, 5}
- **Fark**: A - B = {1, 2}
- **Alt küme**: A ⊆ B (A, B'nin alt kümesi değildir)

java.util.Set Arayüzü Metodları

- **add(E eleman):** Belirtilen elemanı kümeye ekler.
- **remove(Object eleman):** Belirtilen elemanı kümeden çıkarır.
- **contains(Object eleman):** Elemanın kümede olup olmadığını döndürür.
- **size():** Kümenin eleman sayısını döndürür.
- **isEmpty():** Kümenin boş olup olmadığını söyler.
- **clear():** Kümeden tüm elemanları çıkarır.

java.util.Set Arayüzü Uygulamaları

- **HashSet**: Elemanları hash fonksiyonu çıktısına göre bir sırada saklar.
- **LinkedHashSet**: Elemanları kümeye eklenme sırasına göre saklar.
- **TreeSet**: Elemanları belirli sırada saklar. (alfabetik gibi)

Hash Tabanlı Küme

- Ekleme ve arama işlemleri hızlıdır.
- Küme, bir hash tablosu olarak temsil edilir.
- Her bir eleman, hash koduna dayalı olarak saklanır.
- Elemanların konumu hızlı bir şekilde hesaplanabilir.

Ağaç Tabanlı Küme

- Elemanları sıralı bir şekilde saklar.
- Küme, bir ikili arama ağacı olarak temsil edilir.
- Elemanlar sıralı bir şekilde saklanır.
- Arama işlemi O(log n) zaman karmaşıklığına sahiptir.

SON