

## VERİ YAPILARILARI VE ALGORİTMALAR

Graph Traversal  
Breadth First Search (BFS)

# Breadth First Search (BFS)

- Bir çizge ya da ağacı keşfetmek üzere kullanılan bir yöntemdir.
- Bu yaklaşım ile önce bir adım ötede, sonra iki adım ötede olan diğer düğümleri keşfedersiniz.
- BFS bir göletin ortasına taş atmak gibidir. Keşfettiğiniz düğümler taşın merkezinde çevreye doğru yayılan dalgalara benzetilebilir.

# Breadth First Search (BFS)

- BFS, ağaç veri yapısındaki **Level Order Traversal** gibi çalışır.
- Yardımcı veri yapısı olarak kuyrukları (**Queues**) kullanır.
- Aslında **Level Order Traversal** algoritması BFS'den ilham alınarak geliştirilmiştir.

# Breadth First Search (BFS)

## Algoritmasının Takip Ettiği Adımlar

1. Ziyaret edilmemiş komşu düğümü ziyaret et. Bu düğümü ziyaret edilmiş olarak işaretle. Düğümü göster ve kuyruğa ekle.
2. Eğer komşu düğüm yok ise, düğümü kuyruktan çıkar (**DeQueue**)
3. Kuyruk boş kalana kadar 1. ve 2. adımı takip et.

# Breadth First Search (BFS)

## Pseudo Code

```
BFS(input: graph G) {  
  Queue Q;      Integer x, z, y;  
  while (G has an unvisited node x) {  
    visit(x); Enqueue(x,Q);  
    while (Q is not empty){  
      z := Dequeue(Q);  
      for all (unvisited neighbor y of z) {  
        visit(y); Enqueue(y,Q);  
      }  
    }  
  }  
}
```

2

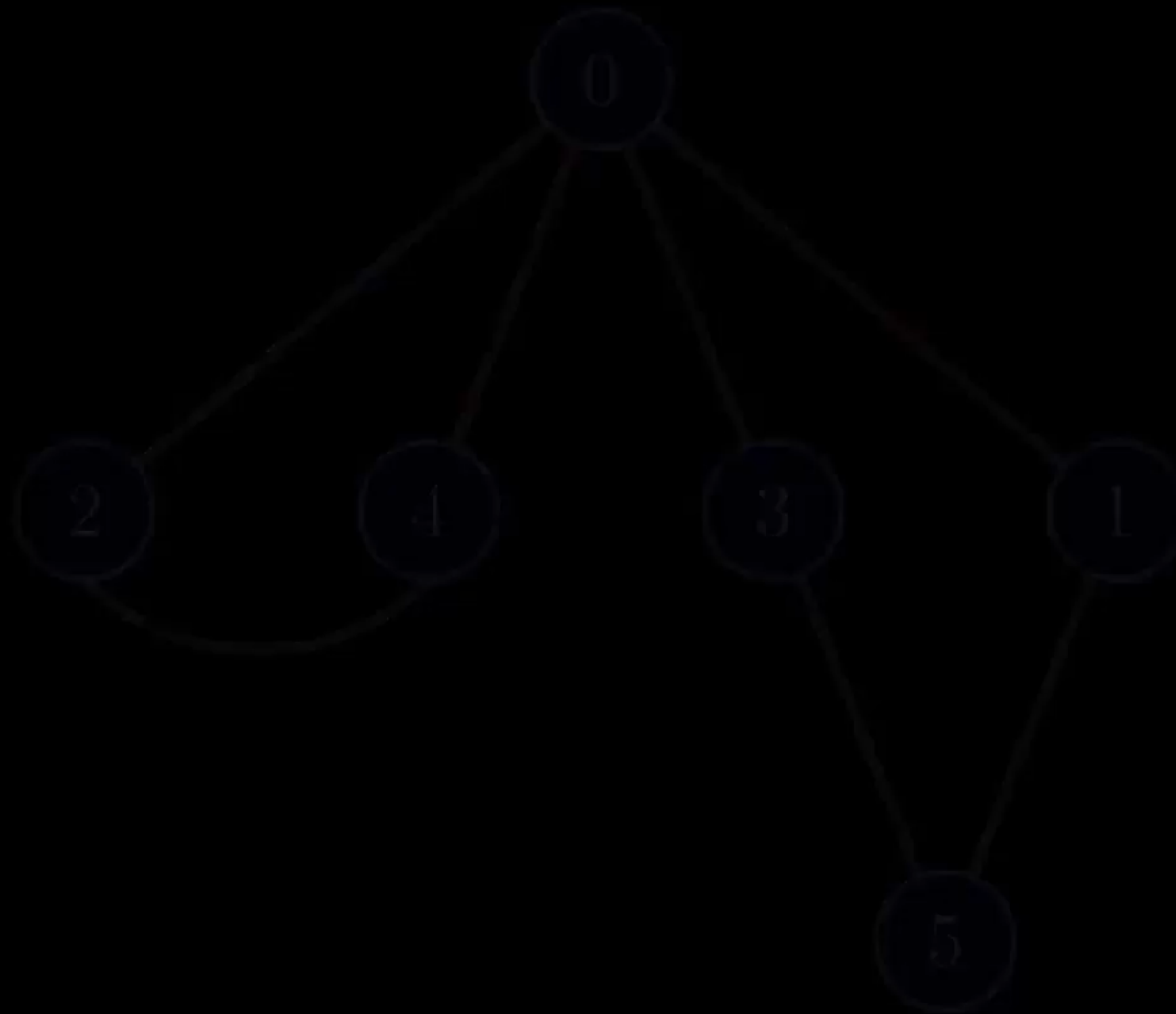
0

1

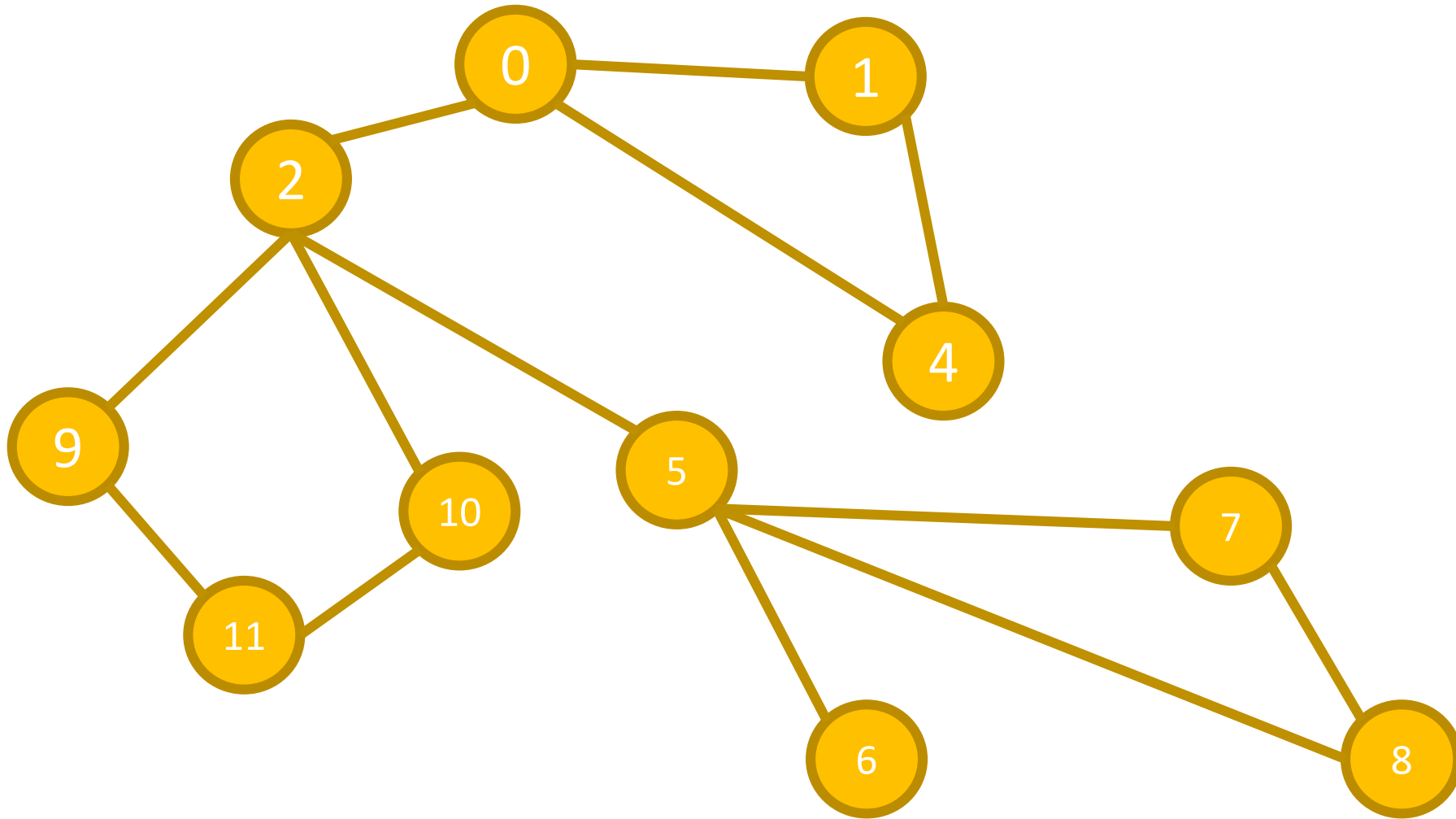
4

3

5

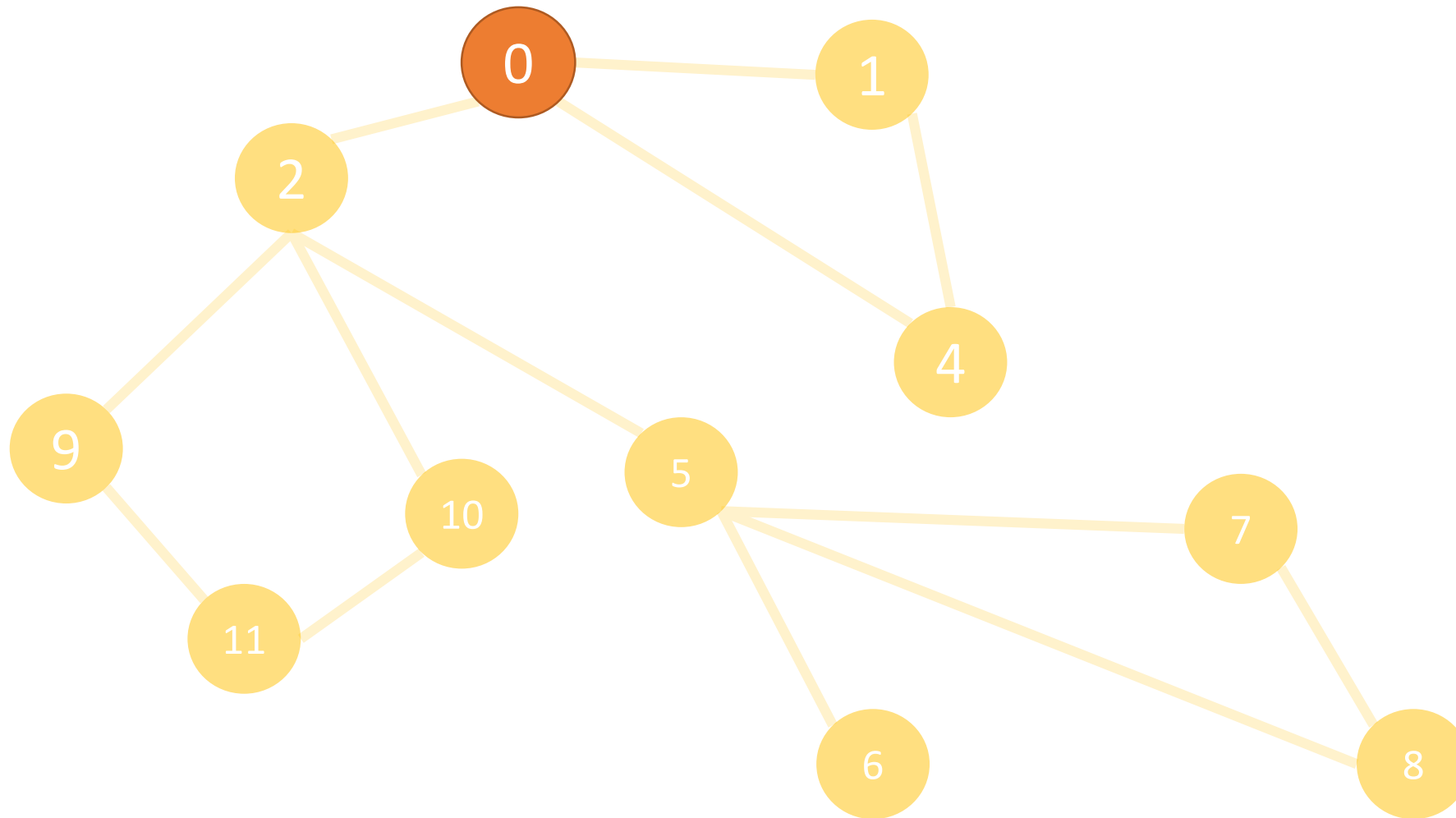


# Breadth First Search (BFS)

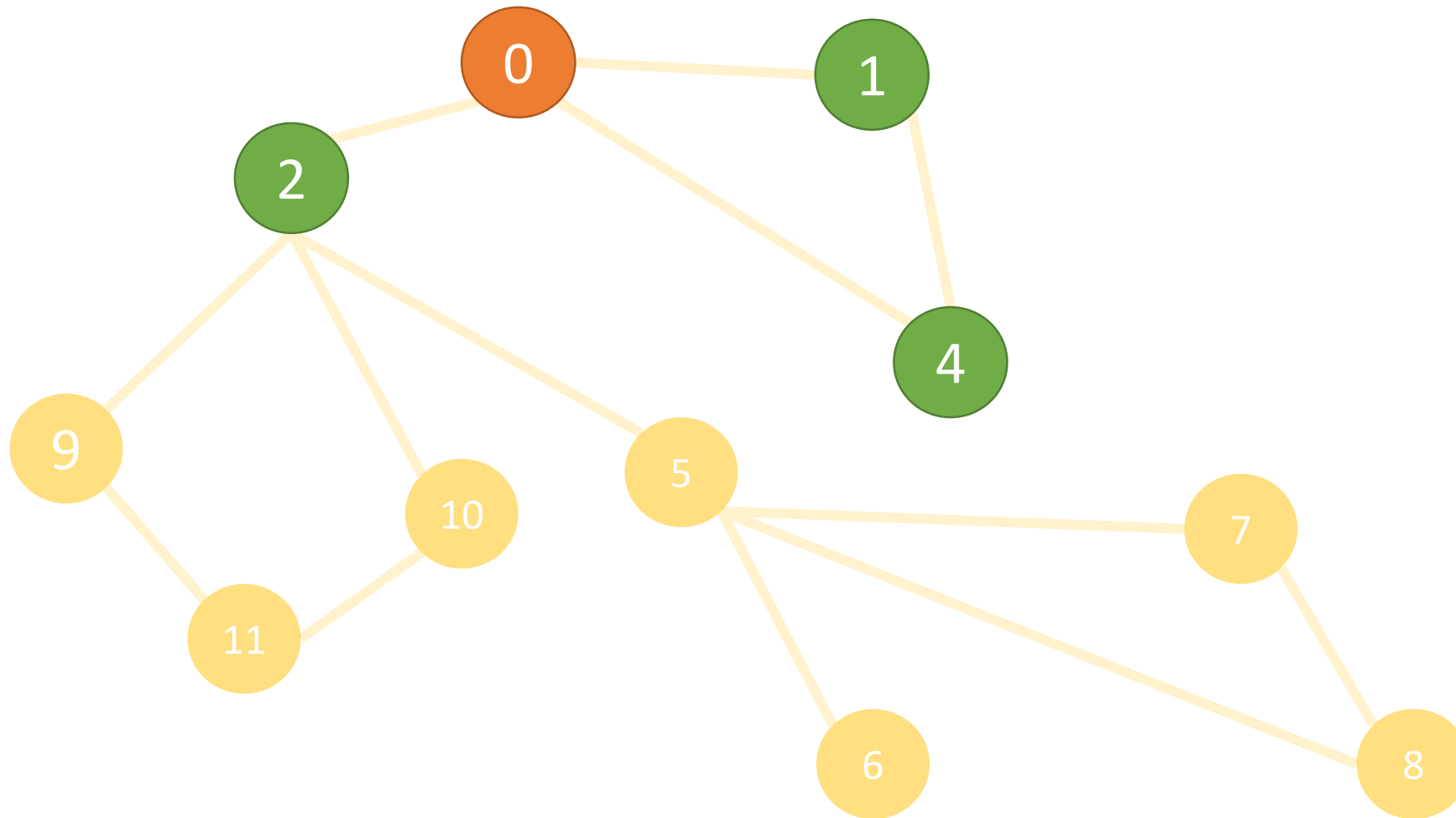




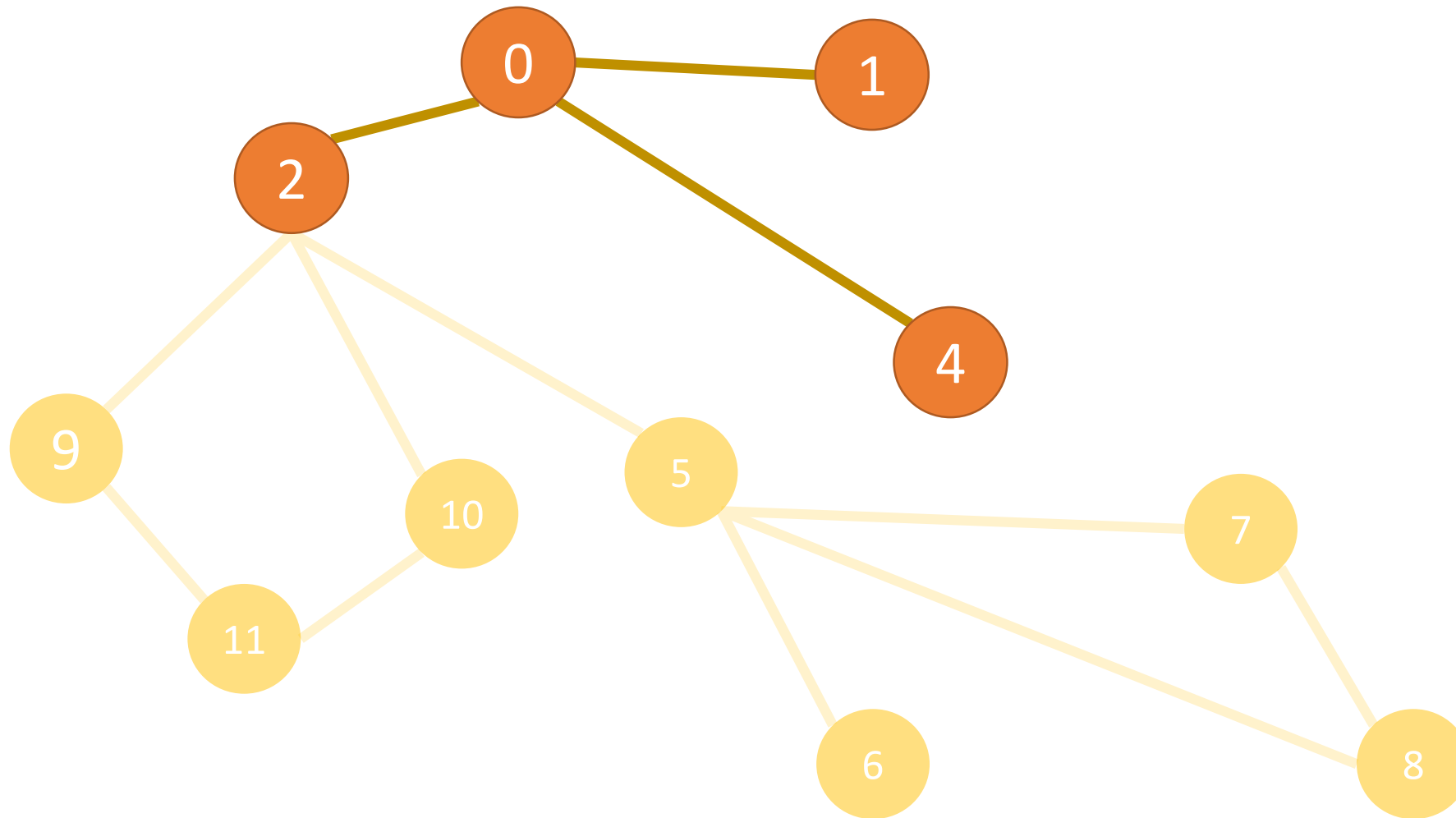
# Breadth First Search (BFS)



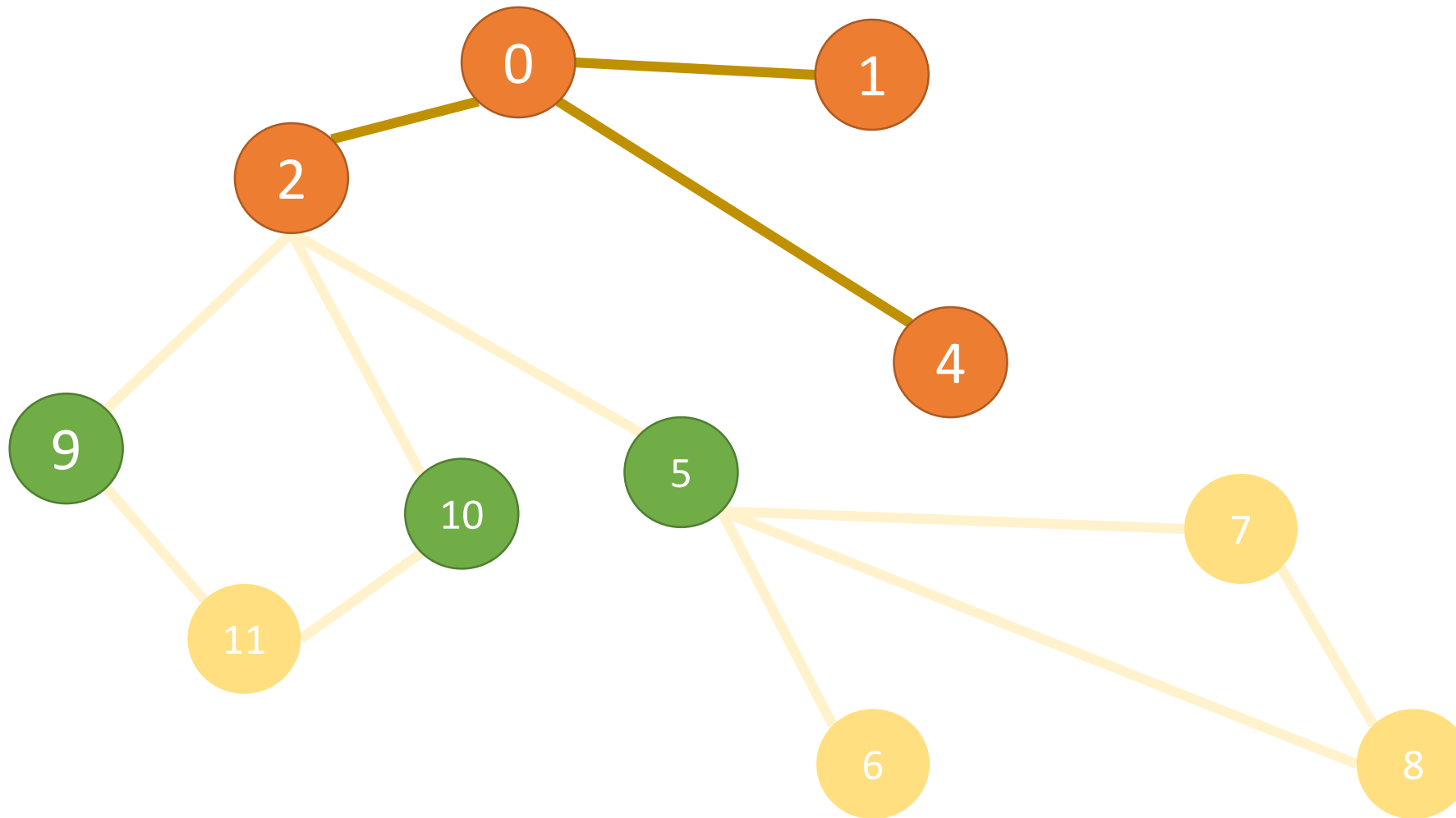
# Breadth First Search (BFS)



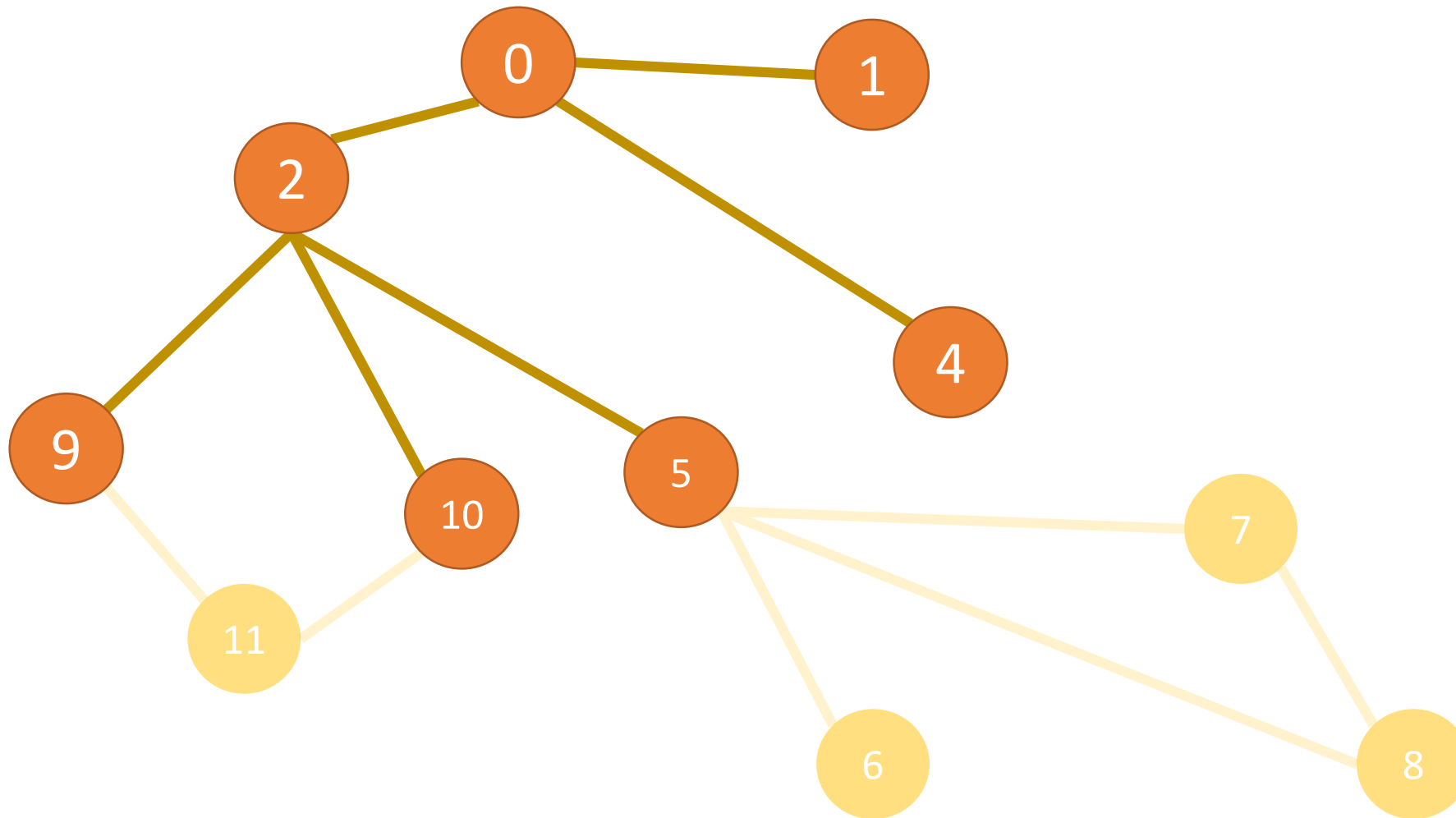
# Breadth First Search (BFS)



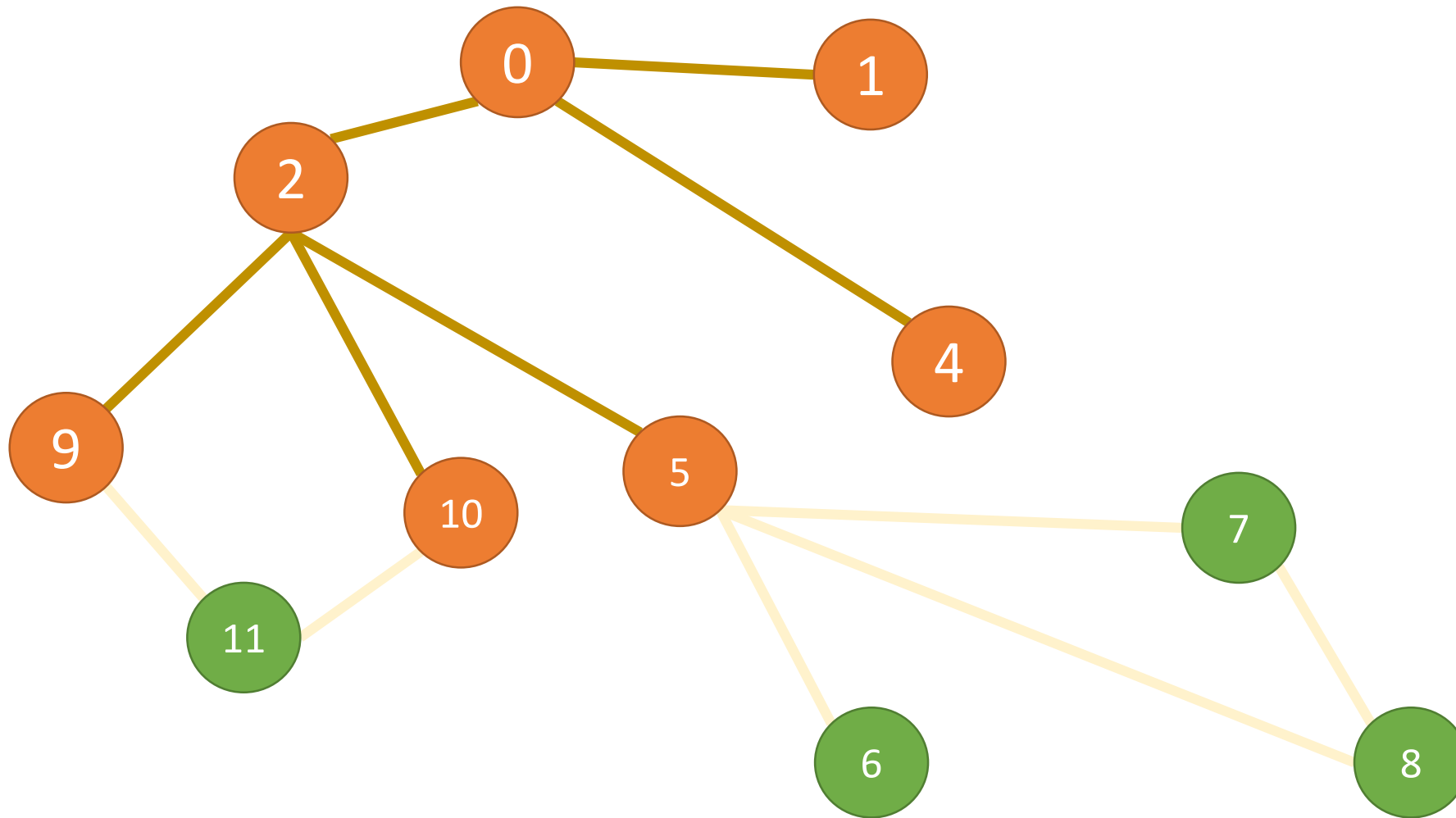
# Breadth First Search (BFS)



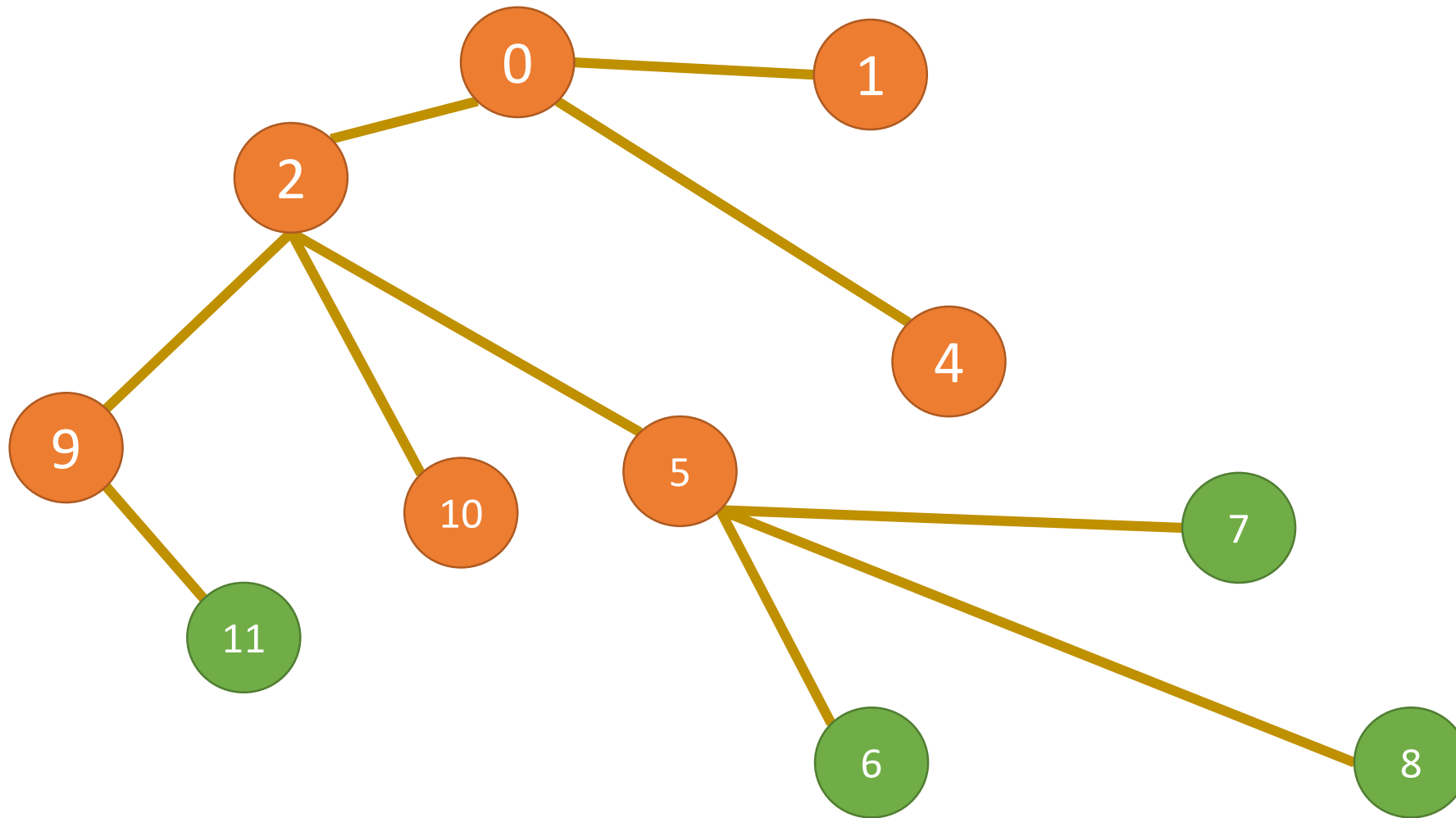
# Breadth First Search (BFS)



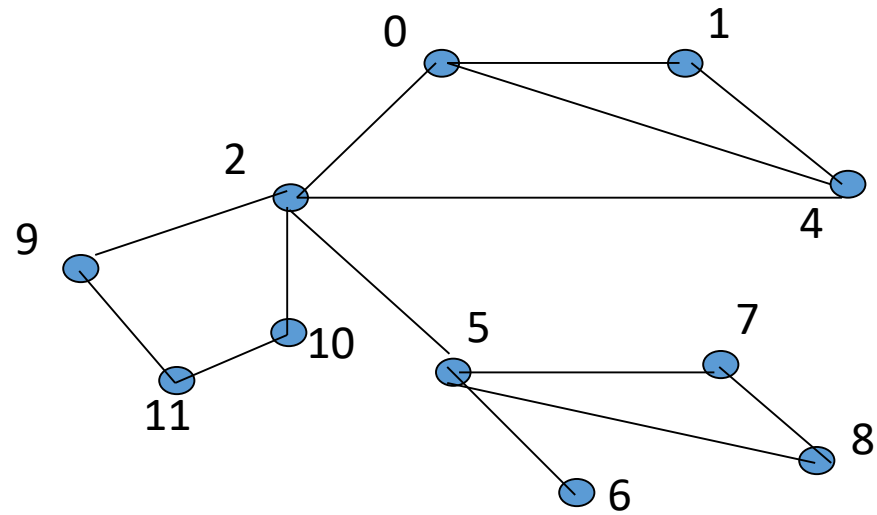
# Breadth First Search (BFS)



# Breadth First Search (BFS)



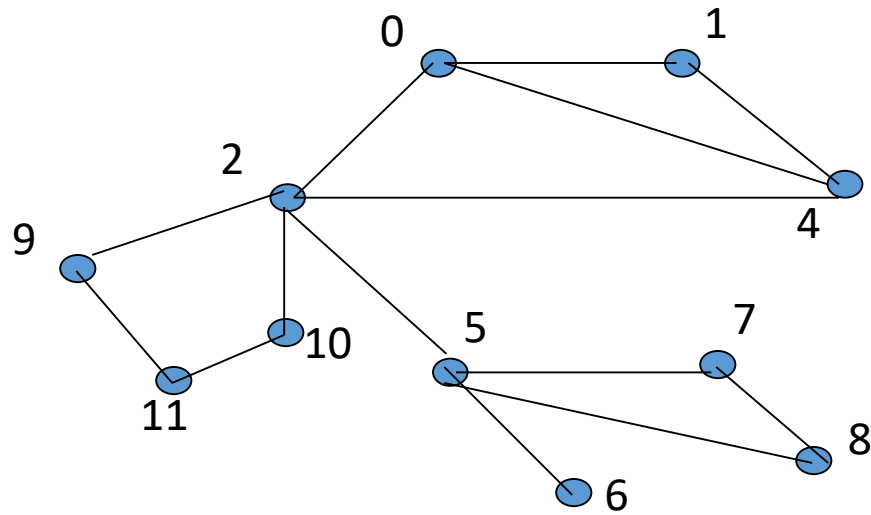
# Breadth First Search (BFS)



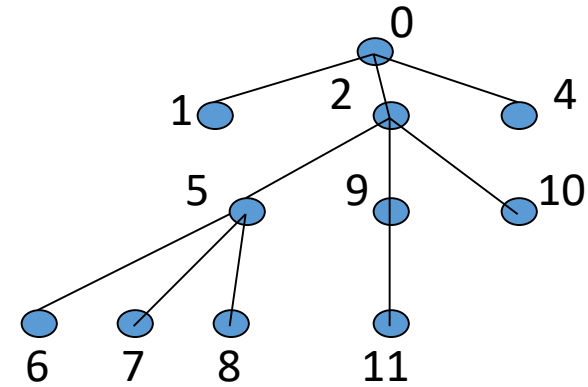
Graph G



# Breadth First Search (BFS)



Graph G



BFS Tree

# Breadth First Search (BFS)

- **Avantajları**

- Bir BFS algoritması başlangıç düğümü ile erişilebilen düğümler arasındaki en kısa yolu bulur.
- DFS algoritmasında en kısa yolu bulma gerekliliği yoktur.

- **Dezavantajları**

- İkili ağaç üzerindeki bir BFS algoritması genellikle DFS algoritmasına kıyasla daha fazla hafızaya ihtiyaç duyar.

# Breadth First Search (BFS)

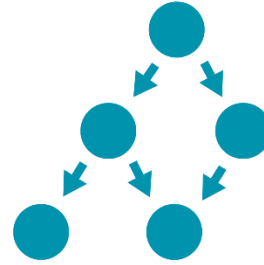
## Uygulamaları (Applications)

- Bir çizgedeki tüm bağlı bileşenleri bulma
- Tek bağlı bileşendeki tüm düğümleri bulma.
- İki düğüm arasındaki en kısa yolu bulma.
- İki-parçalı çizge (bipartite graph) testi

# DFS vs BFS

Uygulamalar	DFS	BFS
Spanning forest, connected components, paths, cycles	+	-
En kısa yol	-	+
Efektif hafıza kullanımı	+	-

- DFS çok daha az hafıza ihtiyacı nedeniyle BFS kıyasla büyük bir avantaja sahiptir; çünkü her seviyedeki çocukların işaretçilerini depolamak zorunda değildir.
- BFS son seviye erişmesi daha uzun zaman alabilir.
- DFS algoritması hedefi daha hızlı bulur.
- DFS daha çok preOrder traversal gibi çalışırken; BFS level-order traversal gibi çalışır.
- Problem tanımı ve veriye bağlı olarak DFS ve BFS üstünlüğü konusu değerlendirilmelidir.



Veri Yapıları ve Algoritmalar

**ZAFER CÖMERT**

Öğretim Üyesi