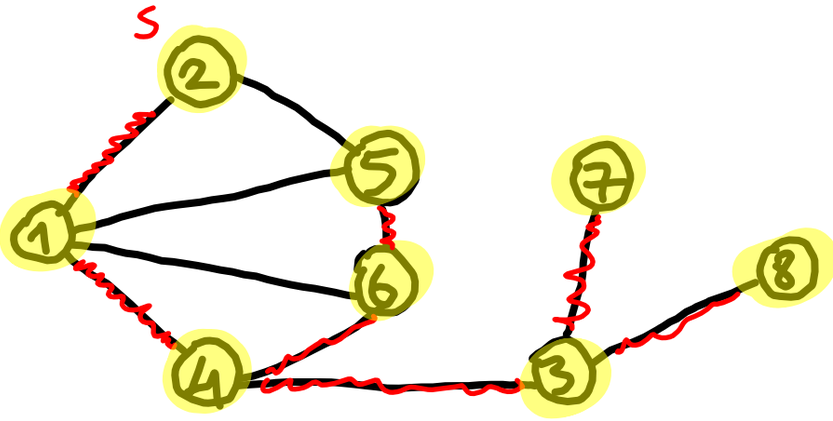
DFS (Depth First Search)
Derinlik Öncelikli Arama

```
DFS(G,s)
    mark(s);
    L = {s};
    while  L ≠ ∅
        u = last(L);
        if ∃ (u,v) such that v is unmarked      ← there exists
            choose (u,v) with v of smallest index;   ← ∈ E
            mark(v);
            L = L ∪ {v};      //  push(L,v)
        else
            L = L \ {u};      //  pop(L)
```

DFS de L için stack (Last in First out) kullanılır.

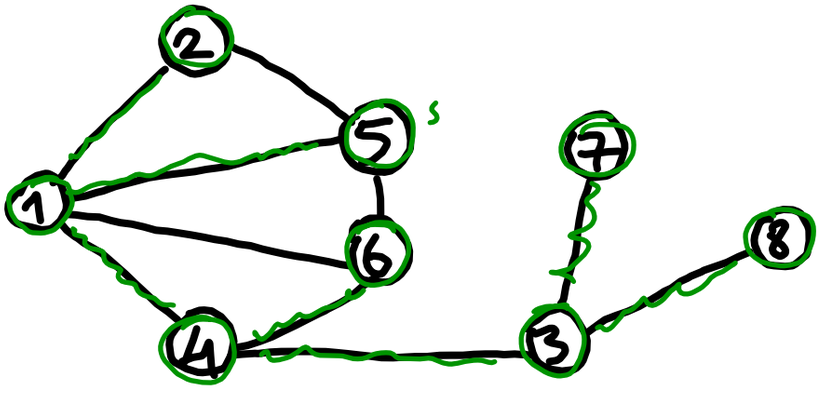
L (stack)	Marked
2	s=2
2, 1 ← top	1
2, 1, 4	4
2, 1, 4, 3 ← top	3
2, 1, 4, 3, 7	7
2, 1, 4, 3	—
2, 1, 4, 3, 8	8
2, 1, 4, 3	—
2, 1, 4	—
2, 1, 4, 6	6
2, 1, 4, 6, 5	5
2, 1, 4, 6	—
2, 1, 4	—
2, 1	—
2	—
∅	—




```
while  L≠∅          ← top
    u= last(L);
    if ∃ (u,v) such that v is unmarked
       choose (u,v) with v of smallest index
       mark(v);
       L= L∪{v};   // push(L,v)
    else
       L= L\{u};   // pop(L)
```

Çalışma Zamanı:

$$T(|V|,|E|) = \Theta(E+V)$$

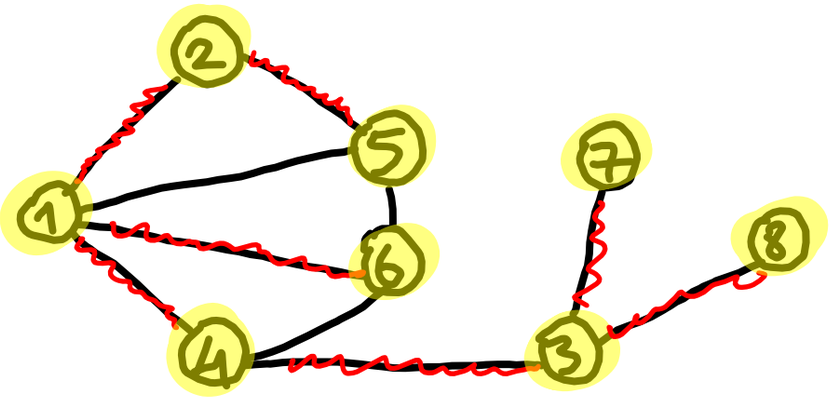


BFS (Breadth First Search)
Genişlik Öncelikli Arama

```
BFS(G,s)
    mark(s).
    L = {s};        // L kuyruktur.
    while  L ≠ ∅
        u= first(L);
        if ∃ (u,v) such that v is unmarked
           choose (u,v) with v of smallest index;
           mark(v);
           L= L U {v};   //  enqueue(L,v)
        else
           L= L \ {u};   //  dequeue(L)
```

L için kuyruk veri yapısı (First in first out) kullanılır.

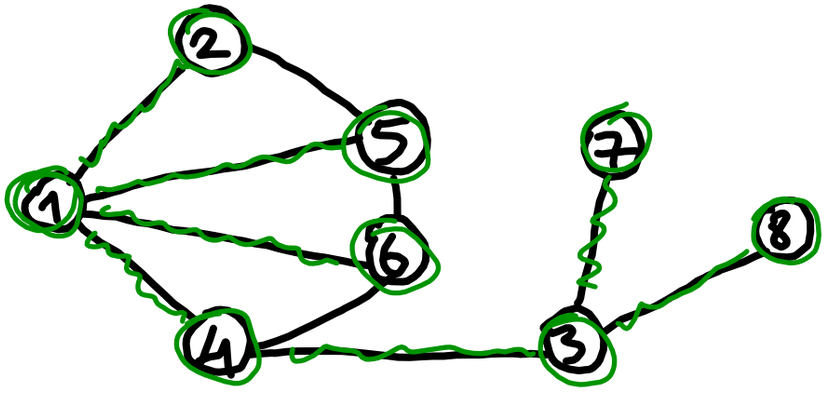
L (kuyruk)	Marked
front ← 2 → rear	s = 2
2, 1	1
2, 1, 5	5
1, 5	—
1, 5, 4	4
1, 5, 4, 6	6
5, 4, 6	—
4, 6	—
4, 6, 3	3
6, 3	—
3	—
3, 7	7
3, 7, 8	8
7, 8	—
8	—
∅	—



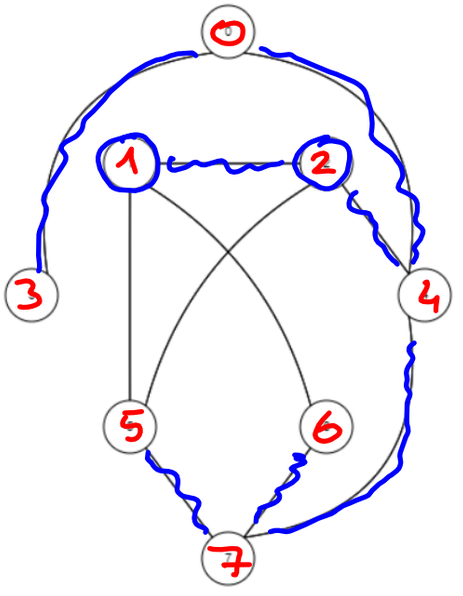

```
while   L ≠ ∅
   u = first(L);
   if ∃ (u,v) such that v is unmarked
      choose (u,v) with v of smallest index;
      mark(v);
      L = L ∪ {v};   //   enqueue(L,v)
   else
      L = L \ {u};   //   dequeue(L)
```

Çalışma Zamanı:

$T(|V|,|E|) = \Theta(E+V)$



DFS : stack



BFS : kuyruk