Discrete Mathematics
LECTURE 14
Spanning Tree

Assistant Professor Gülüzar ÇİT

Outline

- Spanning Tree
- References

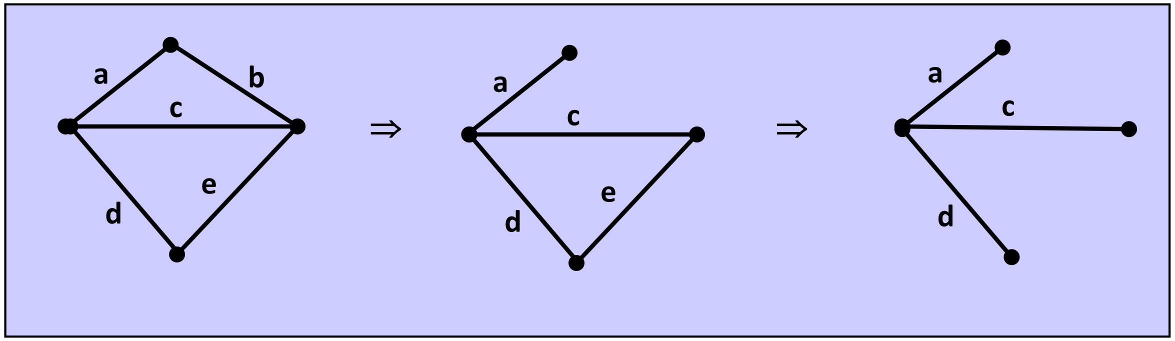
Spanning Tree

- a new tree generated from the nodes and edges of an existing graph
- includes some of the edges and all the nodes of the existing graph

- if the graph is a tree, then the spanning tree is also itself
- a graph will have more than one spanning tree

Spanning Tree...

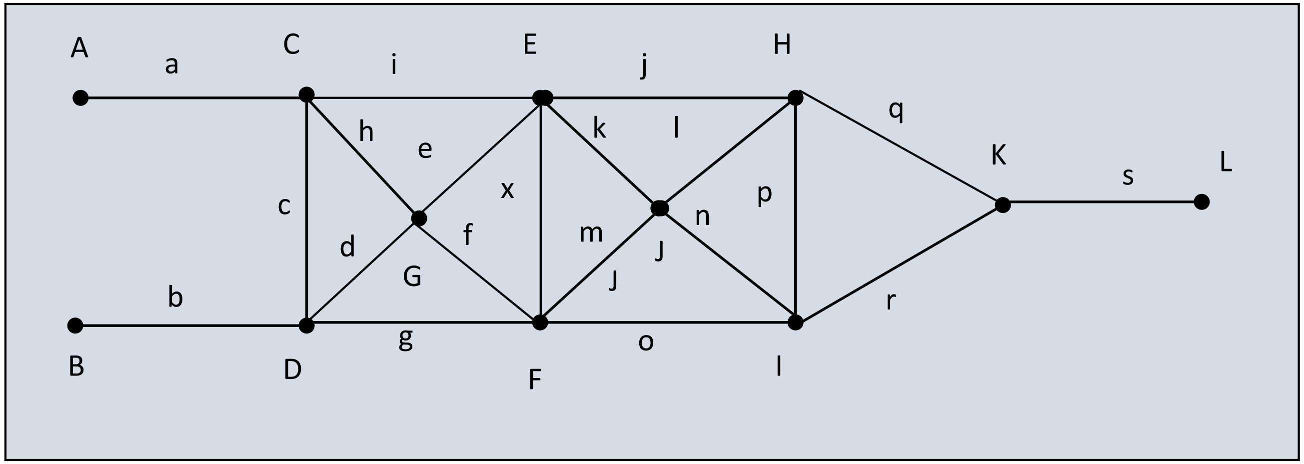
- there are different ways two find the spanning tree of a graph
- one way would be to delete the edges aiming to remove the loops.



- If a connected graph has n nodes and e edges (e ≥ n), we will need to do the edge removal e-n+1 times. Because there are n-1 edges in an n-node tree.

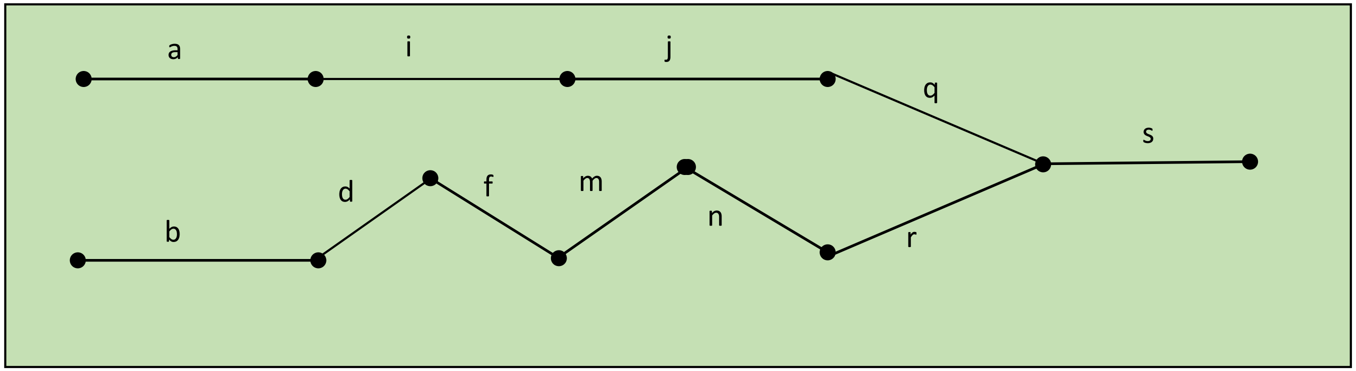
Spanning Tree…

➢ EXAMPLE: Let there be 12 air tanks in a switchyard where there are breakers that open and close with compressed air. All of these tanks have a pipe connection with each other. These connections are shown in the figure. How can the connection be made using the minimum number of pipes?



Spanning Tree...

- **SOLUTION:**
  - The problem is the problem of finding the spanning tree.
  - The solution would be to find a subset of edges that includes all nodes but not include any loop.
  - a-i-j-q-s-r-n-m-f-d-b path is a possible spanning tree that can be constructed from the existing graph since it contains all nodes and does not contain any loop.



Spanning Tree…

- **BFS Spanning Tree Algorithm**
  - starts with a beginning node (S).
  - then finds the neighboring nodes to the beginning node and give it the label 1.
  - Then, neighboring nodes of the node with label 1 are labeled as 2
  - then, continues the process by increasing the label value by one at each step, until there are no unlabeled nodes left in the graph
  - as a result, the edges that lead us to the label k will form the spanning tree.

Spanning Tree…

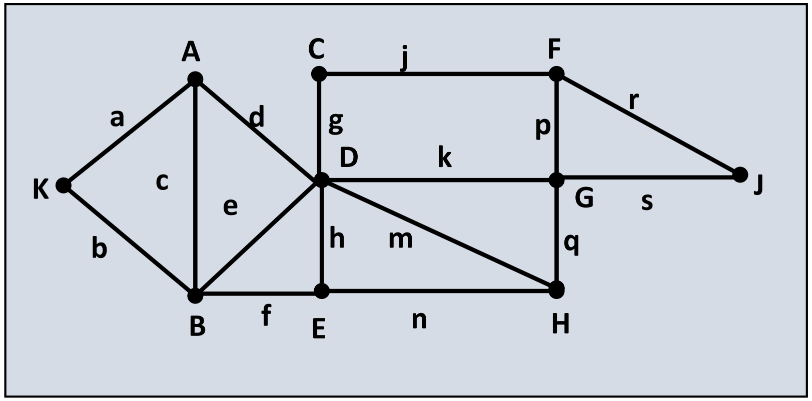
➢BFS Spanning Tree Algorithm…

```
▪ The algorithm finds the spanning tree, if any, for a graph with n nodes.
▪ L is the set of nodes labeled and T is the set of edges connecting the nodes in L.
STEP 1 ( select the starting node)
   STEP 1.1 Select node U, set its label as 0.
   STEP 1.2 L={U}, T={∅} and k=0
STEP 2 ( label nodes)
   WHILE (|L|<n and  ve at least one node in L is adjacent to a node not in L.)
      STEP 2.1 ( increase label)
         k = k + 1
      STEP 2.2 (add one edge to T)
         WHILE (a node V labeled as k-1 in L is adjacent to a node W not in L)
            a) assign label k to node w
            b) add the edge between nodes V and W to T
            c) Add node W to set L
         END WHILE
   END WHILE
STEP 3 (Is there a solution?)
   IF (|L|<n)
      Graph is not connected, therefore there is no spanning tree
   ELSE
      Edges and connected nodes in T form the spanning tree
   END IF
```

Spanning Tree...

➢BFS Spanning Tree Algorithm...

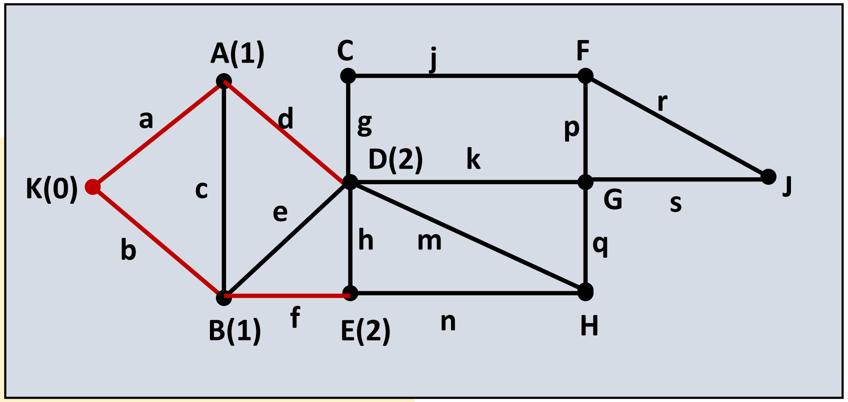
➢Example: Find the spanning tree of the graph by appliying BFS spanning tree algorithm



Spanning Tree…

BFS Spanning Tree Algorithm…

Solution:…

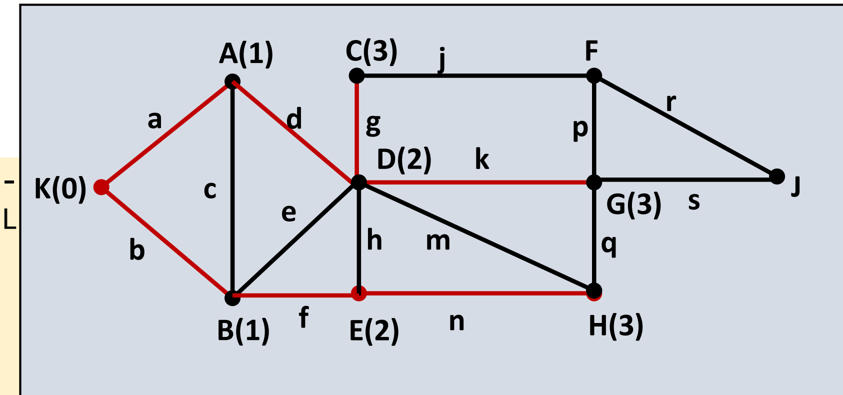



```
ADIM 1 Select K as the starting node
       L = {K}, T = {∅}, k = 0
ADIM 2 (label nodes)
   WHILE (1) (|L|=1<10 and at least one node in
               L is adjacent to a node not in L)
      ADIM 2.1 (increase label) k = 1
      ADIM 2.2 (add one edge to T)
         WHILE (1) (A is adjacent to K)
            A ← 1, L={K,A}, T={a}
         --------------------------------------------------------------
         WHILE (2) (B is adjacent to K)
            B ← 1, L={K,A,B}, T={a,b}
    ------------------------------------------------------------------
    WHILE (2) (|L|=3<10 ve at least one node in L is
                adjacent to a node not in L)
       ADIM 2.1 (increase label) k = 2
       ADIM 2.2
          WHILE (1) (D is adjacent to A)
             D ← 2, L={K,A,B,D}, T={a,b,d}
          -------------------------------------------------------------
          WHILE (2) (E is adjacent to B)
             E ← 2, L={K,A,B,D,E}, T={a,b,d,f}
    ------------------------------------------------------------------
```

Spanning Tree…

BFS Spanning Tree Algorithm…

Solution:…

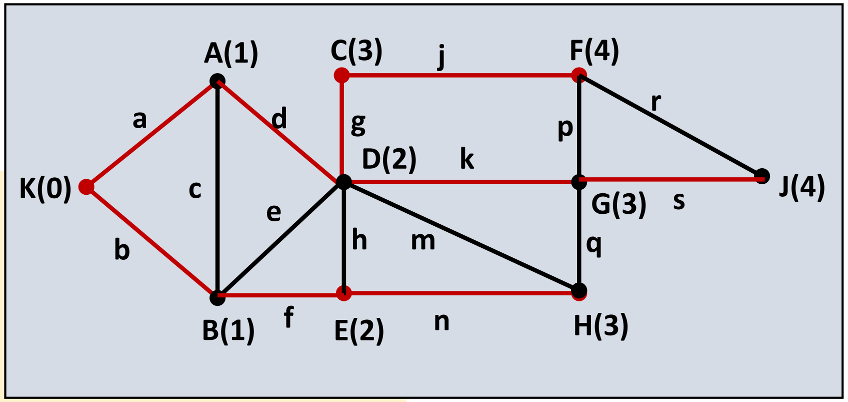



```
-------------------------------------------
WHILE (3) (|L|=5<10 ve at least one node in L
           is adjacent to a node not in L)
  ADIM 2.1 (increase label) k = 3
  ADIM 2.2
     WHILE (1) (C is adjacent to D)
        C ← 3, L={K,A,B,D,E,C}, T={a,b,d,f,g}
     -------------------------------------------------------------
     WHILE (2) (G is adjacent to D)
        G ← 3, L={K,A,B,D,E,C,G}, T={a,b,d,f,g,k}
     -------------------------------------------------------------
     WHILE (3) (H is adjacent to D)
        H ← 3, L={K,A,B,D,E,C,G,H}, T={a,b,d,f,g,k,n}
-------------------------------------------------------------------
```

Spanning Tree…

BFS Spanning Tree Algorithm…

Solution:…

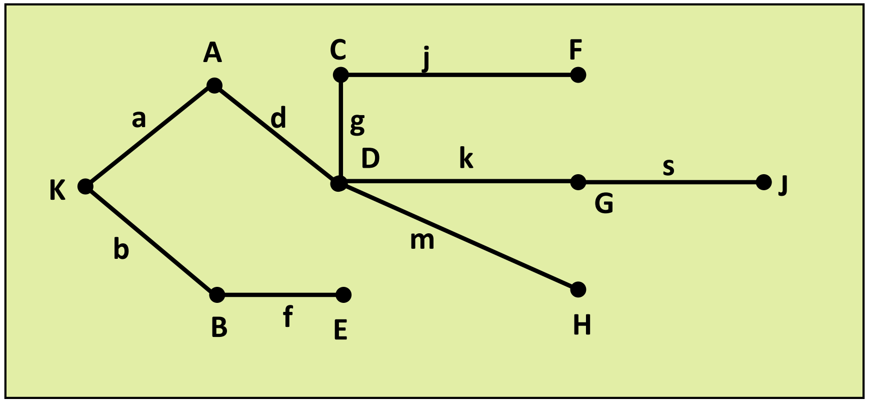



```
 -----------------------------------------------
 WHILE (4) (|L|=8<10 ve at least one node in L
             is adjacent to a node not in L)
    ADIM 2.1 (increase label) k = 4
    ADIM 2.2
       WHILE (1) (F is adjacent to C)
          C ← 4, L={K,A,B,D,E,C,G,H,F}, T={a,b,d,f,g,k,n,j}
       --------------------------------------------------------------
       WHILE (2) (J is adjacent to G)
          J ← 4, L={K,A,B,D,E,C,G,H,F,J}, T={a,b,d,f,g,k,n,j,s}
 --------------------------------------------------------------------
 WHILE (5) (|L|=10)
 END WHILE
 --------------------------------------------------------------------
STEP 3 (Is there a solution?)
   Spanning tree is T={a,b,d,f,g,k,n,j,s}
```

Spanning Tree…

➢BFS Spanning Tree Algorithm…

➢Solution:…



Spanning Tree…

➢ BFS Spanning Tree Algorithm…

➢ Example: Find the spanning tree of graph F whose adjacency list is given below using the BFS spanning tree algorithm

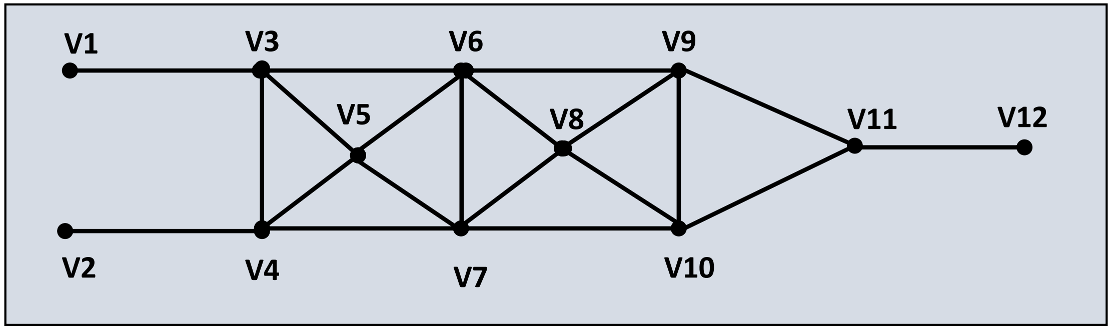
```
V1 → V3
V2 → V4
V3 → V1,V4,V5,V6
V4 → V2,V3,V5,V7
V5 → V3,V4,V6,V7
V6 → V3,V5,V7,V8,V9
V7 → V4,V5,V6,V8,V10
V8 → V6,V7,V9,V10
V9 → V6,V8,V10,V11
V10 → V7,V8,V9,V11
V11 → V9,V10,V12
V12 → V11
```

Spanning Tree…

➢ BFS Spanning Tree Algorithm…

➢ Solution:

```
V1 → V3
V2 → V4
V3 → V1,V4,V5,V6
V4 → V2,V3,V5,V7
V5 → V3,V4,V6,V7
V6 → V3,V5,V7,V8,V9
V7 → V4,V5,V6,V8,V10
V8 → V6,V7,V9,V10
V9 → V6,V8,V10,V11
V10 → V7,V8,V9,V11
V11 → V9,V10,V12
V12 → V11
```



Spanning Tree…

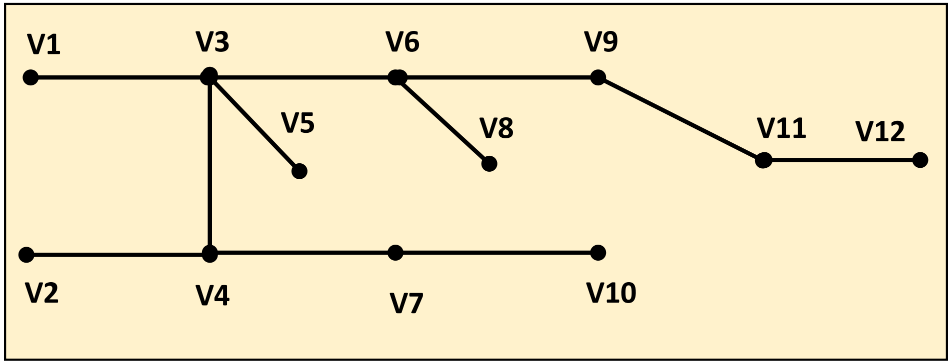
BFS Spanning Tree Algorithm…

Solution:…

```
L={V1 | V3 | V4 V5 V6 | V2 V7 V8 V9 | V10 V11 | V12 }
T={ (V1 V3), (V3 V4), (V3 V5), (V3 V6), (V4 V2), (V4 V7),
(V6 V8), (V6 V9),(V7 V10), (V9 V11),(V11 V12)}
```

```
V1 → V3
V2 → V4
V3 → V1,V4,V5,V6
V4 → V2,V3,V5,V7
V5 → V3,V4,V6,V7
V6 → V3,V5,V7,V8,V9
V7 → V4,V5,V6,V8,V10
V8 → V6,V7,V9,V10
V9 → V6,V8,V10,V11
V10 → V7,V8,V9,V11
V11 → V9,V10,V12
V12 → V11
```

k=	1	2	3	4	5
V1	0	X	X	X	X
V2	-	-	3	X	X
V3	1	X	X	X	X
V4	-	2	X	X	X
V5	-	2	X	X	X
V6	-	2	X	X	X
V7	-	-	3	X	X
V8	-	-	3	X	X
V9	-	-	3	X	X
V10	-	-	-	4	X
V11	-	-	-	4	X
V12	-	-	-	-	5



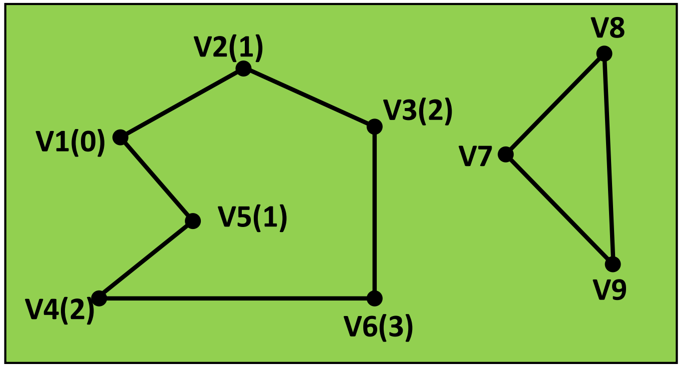
Kapsama Ağaçları…

BFS Spanning Tree Algorithm…

- **Example:** Find the spanning tree of the graph whose adjacency list is given on the right

V1	:	V2,V5
V2	:	V1,V3
V3	:	V2,V6
V4	:	V5,V6
V5	:	V1,V4
V6	:	V3,V4
V7	:	V8,V9
V8	:	V7,V9
V9	:	V7,V8

- **Solution:**



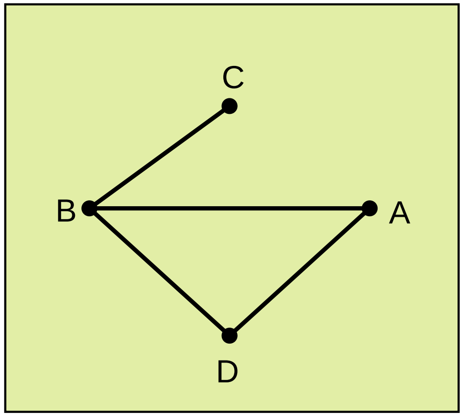
- There is no spanning tree, because the grapgh is not connected ✓

Spanning Tree…

- **Depth First Search Spanning Tree Algorithm**
  - another algorithm to find the spanning tree of a graph
  - Nodes are labeled with consecutive integers
    - The basic idea here is to tag a node v and then immediately tag another node from its neighbors.
    - If the node w is adjacent to the node v, it is given the next tag and
    - immediately starts searching for a node adjacent to the node w.
    - If node v has no unlabeled neighbor node, this situation continues step by step backwards from the way we came to node v and
    - if we find an unlabeled node u adjacent to the node every time we go back, this node search process is continued starting from node u.

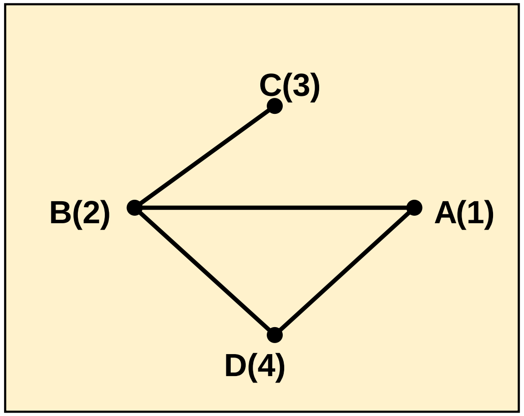
Spanning Tree…

- **Depth First Search Spanning Tree Algorithm…**
  - Example: Find the spanning tree applying DFS for the graph given below



Spanning Tree…

- **Depth First Search Spanning Tree Algorithm…**
  - Solution:
    - Let A be the starting node, its label will be 1.
    - Let A's neighbors be B and D and choose B randomly.
    - The label of node B will be 2.Let's choose B's neighbors C and D, C.
    - C has no other neighbors, so we go back and look at B's neighbors.
    - Since node D is unlabeled, it will get the next label i.e. 4.



Spanning Tree…

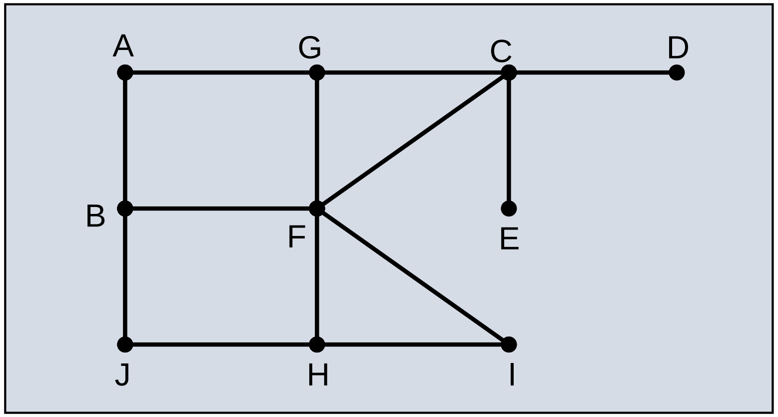
Depth First Search Spanning Tree Algorithm…

```
▪ The algorithm finds the spanning tree, if any, for a graph with n nodes.
▪ L is the set of nodes labeled and T is the set of edges connecting the nodes in L
  and the prior of node Y is the node used to label Y in L.
STEP 1 (Label starting node)
   Select node U, set its label as 0 and set its prior as NULL.
   L={U}, T={∅}, k = 2, X=U
STEP 2 (Label other nodes)
   REPEAT
      STEP 2.1 (Label an adjacent node to X)
      WHILE (X has an adjacent node Y that is not in L)
         add edge {X,Y} kenarını to T, add node Y to L
         assign label k to Y and set its prior as X
         (let node X point to node Y)
         k = k + 1
      END WHILE
      STEP 2.2 (back step)
         replace X with predecessor of X
   UNTIL (all nodes in g are in L or X=NULL)
STEP 3 (Is there a solution?)
   IF (all nodes in L are in g)
      Edges in T and relevant noders form the spanning tree
   ELSE
       There is not a spanning tree of graph g and graph is not connected
   END IF
```

Spanning Tree…

➢ Depth First Search Spanning Tree Algorithm…

- ➢ Example: Find the spanning tree applying DFS spanning tree algorithm for the graph given below



Spanning Tree…

Depth First Search Spanning Tree Algorithm…

Solution:

```
ADIM 1 Başlangıç düğümü A, L={A}, T={∅}, k=2, X=A
ADIM 2
   REPEAT (1)
      ADIM 2.1
      WHILE (1)
      A'ya komşu B,G var. B'yi seçelim.
      T={(A,B)}, L={A,B}, B→B(2), B←A, k=3, X⇒B
      ----------------------------------------------------------------------
      WHILE (2)
      B'ye komşu F,J var. F'yi seçelim.
      T={(A,B),(B,F)}, L={A,B,F}, F→F(3), F←B, k=3, X⇒F
      ----------------------------------------------------------------------
      WHILE (3)
      F'ye komşu C,G,I,H var. C'yi seçelim.
      T={(A,B),(B,F),(F,C)}, L={A,B,F,C}, C→C(4), C←F, k=5, X⇒C
      ----------------------------------------------------------------------
      WHILE (4)
      C'ye komşu D,E ve G var. D'yi seçelim.
      T={(A,B),(B,F),(F,C),(C,D)}, L={A,B,F,C,D}, D→D(5), D←C, k=6, X⇒D
      ----------------------------------------------------------------------
      WHILE (5)
      D'ye komşu yok
      END WHILE
      ADIM 2.2 X=C
```

Spanning Tree…

➢ Depth First Search Spanning Tree Algorithm…

➢ Solution:…

```
REPEAT (2)
    ADIM 2.1
    WHILE (1)
    C’ye komşu E,G var. E’yi seçelim.
    T={(A,B),(B,F),(F,C),(C,D),(C,E)},
    L={A,B,F,C,D,E}, E→E(6), E←C, k=7, X⇒E
    ----------------------------------------------------------------------
    WHILE (2)
    E’ye komşu yok.
    END WHILE
    ----------------------------------------------------------------------
    ADIM 2.2  X=C
 -------------------------------------------------------------------------
 REPEAT (3)
    ADIM 2.1
    WHILE (1)
    C’ye komşu G var. G’yi seçelim.
    T={(A,B),(B,F),(F,C),(C,D),(C,E),(C,G)}
    L={A,B,F,C,D,E,G}, G→G(7), G←C, k=8, X⇒G
    ----------------------------------------------------------------------
    WHILE (2)
    G’ye komşu yok.
    END WHILE
    ----------------------------------------------------------------------
    ADIM 2.2  X=C
```

Spanning Tree…

➢Depth First Search Spanning Tree Algorithm…

➢Solution:…

```
REPEAT (4)
    ADIM 2.1
    WHILE (1)
    C’ye komşu yok.
    END WHILE
    -----------------------------------------------------------------------
    ADIM 2.2  X=F
 ---------------------------------------------------------------------------
 REPEAT (5)
    ADIM 2.1
    WHILE (1)
    F’ye komşu H,I var. H’yi seçelim.
    T={(A,B),(B,F),(F,C),(C,D),(C,E),(C,G),(F,H)}
    L={A,B,F,C,D,E,G,H}, H→H(8), H←F, k=9, X⇒H
    -----------------------------------------------------------------------
    WHILE (2)
    H’ye komşu I,J var. I’yi seçelim.
    T={(A,B),(B,F),(F,C),(C,D),(C,E),(C,G),(F,H),(H,I)}
    L={A,B,F,C,D,E,G,H,I}, I→I(9), I←H, k=10, X⇒I
    -----------------------------------------------------------------------
    WHILE (3)
    I’ya komşu yok.
    END WHILE
    -----------------------------------------------------------------------
    ADIM 2.2  X=H
```

Spanning Tree…

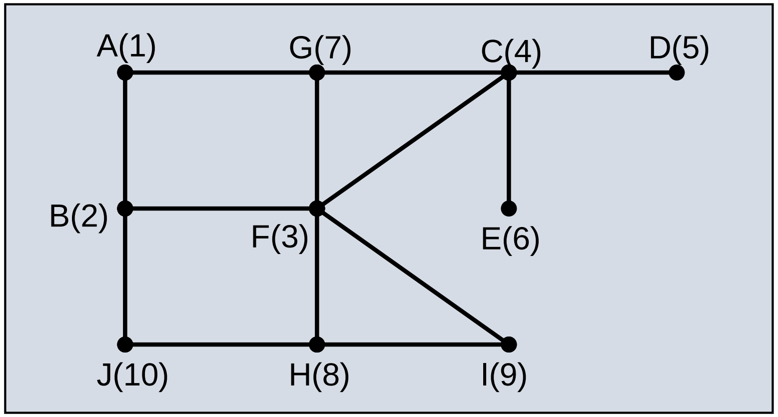
➢Depth First Search Spanning Tree Algorithm…

➢Solution:…

```
REPEAT (6)
   ADIM 2.1
   WHILE (1)
   H’ye komşu J var.
   T={(A,B),(B,F),(F,C),(C,D),(C,E),(C,G),(F,H),(H,I),(H,J)}
   L={A,B,F,C,D,E,G,H,I,J}, J→J(10), J←H, k=11, X⇒J
   ----------------------------------------------------------------------
   WHILE (2)
   J’ye komşu yok.
   END WHILE
   ----------------------------------------------------------------------
   ADIM 2.2  X=H
UNTIL (tüm düğümler L’de)
-------------------------------------------------------------------------
ADIM 3
   IF (L’de tüm düğümler var)
   T={(A,B),(B,F),(F,C),(C,D),(C,E),(C,G),(F,H),(H,I),(H,J)}
   kapsama ağacıdır.
```

Spanning Tree...

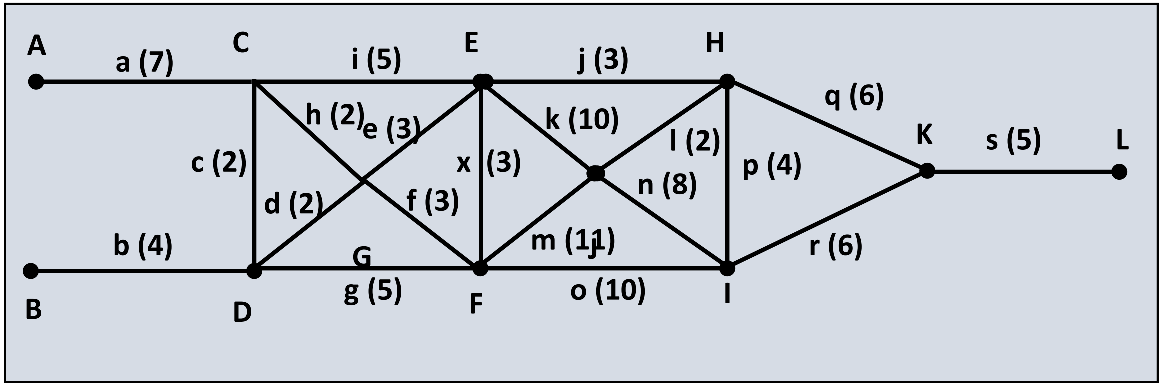
- **Depth First Search Spanning Tree Algorithm...**
  - Solution:...



Spanning Tree…

➢Minimum Spanning Tree

- ➢Spanning tree with minimum total weight
- ➢Calculated for weighted graphs



Spanning Tree...

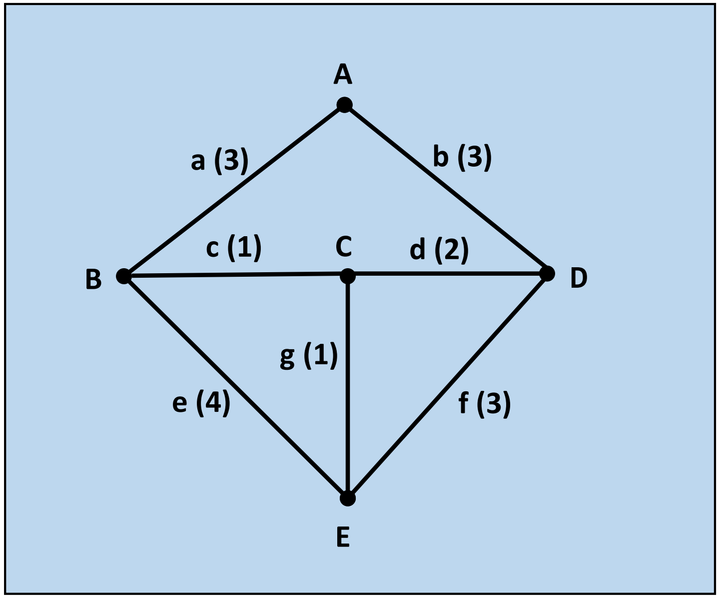
Prim Algorithm

- one of the algorithms that finds the minimal spanning tree.
- a node is selected and a minimum weighted edge is added to it. Then, the tree is started to be created by selecting the minimum edge connected to it.

```
▪ The algorithm finds the minimal spanning tree of a weighted graph with n nodes, if any.
▪ T is the set of edges forming the tree, L is the nodes related to the edges in T.
STEP 1 ( Select a node U) L={U}, T={∅}
STEP 2
   WHILE (|L|<n and there is at least one edge between the node(s) in L and a node not in L)
      STEP 2.1 Select the edge with the least weight among these edges.
      STEP 2.2 Add edge to T
      STEP 2.3 Add the other node connected to this edge to L
   END WHILE
STEP 3 (Is there a solution?)
   IF (|L|<n)
       Graph is not connected, therefore there is no minimal spanning tree
   ELSE
       Edges and connected nodes in T form the minimal spanning tree
   END IF
```

Spanning Tree…

- **Prim Algorithm…**
  - Example: Find the minimal spanning tree for the graph given below

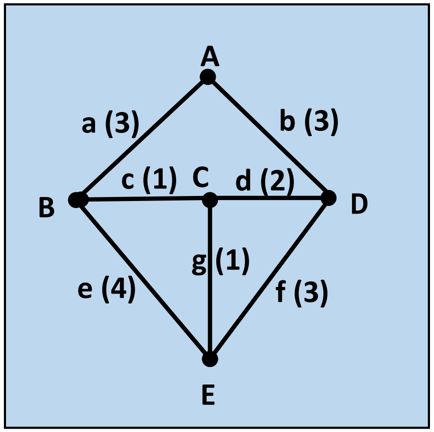


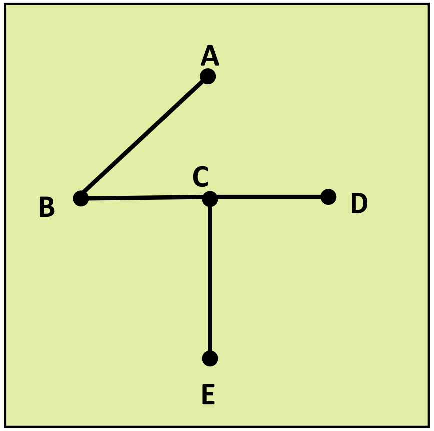
Spanning Tree…

➢ Prim Algorithm…

➢ Solution:

```
STEP 1 ( select node A) L={A}, T={∅}
STEP 2
   WHILE (1) (|1|<5 and edges a and b are connected  to node A)
      The least weighted edge is a and the other node connected to a is node B
      L={A,B}, T={a}
   -------------------------------------------------------------
   WHILE (2) (|2|<5 and edges b,e,c are connected to nodes A and B
      The least weighted edge is c and the other node connected to c is node C
      L={A,B,C}, T={a,c}
   -------------------------------------------------------------
   WHILE (3) (|3|<5 ve edges e,g,d,b are connedted to nodes A,B and C)
      The least weighted edge is g and the other node connected to g is node E
      L={A,B,C,E}, T={a,c,g}
   -------------------------------------------------------------
   WHILE (4) (|4|<5 and edges f,b,d are connected to nodes A,B,C and E)
      The least weighted edge is d and the other node connected to d is node D
      L={A,B,C,E,D}, T={a,c,g,d}
   -------------------------------------------------------------
    WHILE (5) (|L|=5)
    END WHILE
 ----------------------------------------------------------------
STEP 3 T={a,c,g,d} creates the minimum spanning tree.
```





Spanning Tree…

➢Kruskal Algorithm

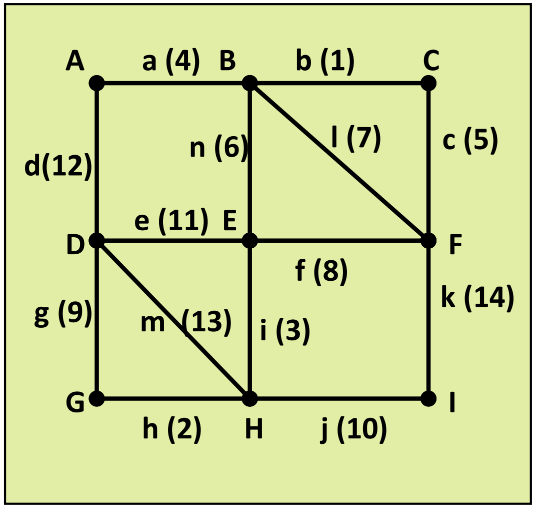
➢Minimal kapsama ağacını bulan algoritmalardan bir diğeridir.

```
▪ The algorithm finds the minimal spanning tree of a weighted graph with n nodes, if any.
▪ T and G are the set of edges of the graph
STEP 1 S={set of all edges of the graph}, T={∅}
STEP 2 (expand T)
   WHILE (|T|<n-1 and S≠{∅})
      STEP 2.1 Select the least weighted edge e in S
      STEP 2.2 If edge e does not create a loop with the other edge in T, add edge e to T
      STEP 2.3 subtract edge e from S
   END WHILE
ADIM 3 (Is there a solution?)
   IF (|T|<n-1)
      Graph is not connected, therefore there is no minimal spanning tree
   ELSE
      Edges and connected nodes in T form the minimal spanning tree
   END IF
```

Spanning Tree…

➢Kruskal Algorithm…

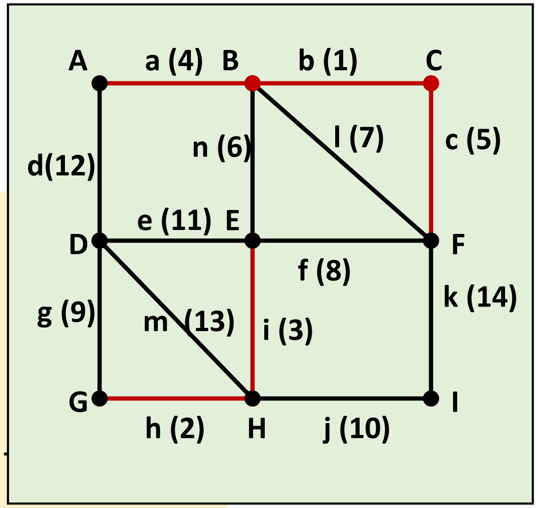
➢Example: Find the minimal spanning tree applying Kruskal algorithm for the graph given below



Spanning Tree…

➢Kruskal Algorithm…

➢Solution:

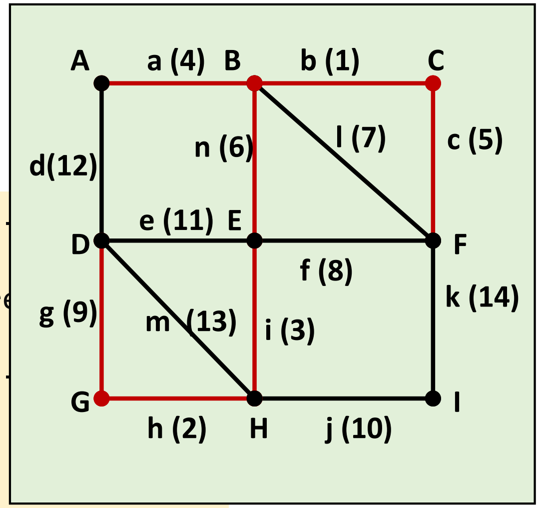



```
STEP 1 T={∅}, S={a,b,c,d,e,f,g,h,i,j,k,l,m,n}
STEP 2
  WHILE (1) (|T|<8 and S={∅})
     the least weighted edge in S is b and
     this edge does not create a loop
     T={b}, S={a,c,d,e,f,g,h,i,j,k,l,m,n}
  -------------------------------------------------------------------------
  WHILE (2) (|T|=1<8 and S={∅})
     the least weighted edge in S is h and this edge does not create a loop
     T={b,h}, S={a,c,d,e,f,g,i,j,k,l,m,n}
  -------------------------------------------------------------------------
  WHILE (3) (|T|=2<8 and S={∅})
     the least weighted edge in S is i and this edge does not create a loop
     T={b,h,i}, S={a,c,d,e,f,g,j,k,l,m,n}
  -------------------------------------------------------------------------
  WHILE (4) (|T|=3<8 and S={∅})
     the least weighted edge in S is a and this edge does not create a loop
     T={b,h,i,a}, S={c,d,e,f,g,j,k,l,m,n}
  -------------------------------------------------------------------------
  WHILE (5) (|T|=4<8 and S={∅})
     the least weighted edge in S is c and this edge does not create a loop
     T={b,h,i,a,c}, S={d,e,f,g,j,k,l,m,n}
```

Spanning Tree…

➢Kruskal Algorithm…

➢Solution:



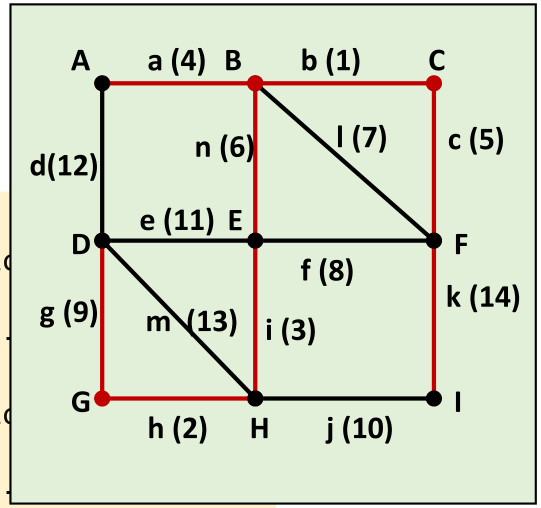

```
-----------------------------------------------------------------------
WHILE (5) (|T|=4<8 and S={∅})
   the least weighted edge in S is n and this edge does not cre
   T={b,h,i,a,c,n}, S={d,e,f,g,j,k,l,m}
-----------------------------------------------------------------------
WHILE (7) (|T|=6<8 and S={∅})
   the least weighted edge in S is l and
   this edge creates a loop
   S={d,e,f,g,j,k,m}
-------------------------------------------------------------------------
WHILE (8) (|T|=6<8 and S={∅})
   the least weighted edge in S is f and this edge creates a loop
   S={d,e,g,j,k,m}
-------------------------------------------------------------------------
WHILE (9) (|T|=6<8 and S={∅})
   the least weighted edge in S is g and this edge does not create a loop
   T={b,h,i,a,c,n,g}, S={d,e,j,k,m}
------------------------------------------------------------------------
WHILE (10) (|T|=7<8 and S={∅})
   the least weighted edge in S is j and this edge creates a loop
   S={d,j,k,m}
-------------------------------------------------------------------------
```

Spanning Tree…

➢Kruskal Algorithm…

➢Solution:

```
WHILE (10) (|T|=7<8 and S={∅})
   the least weighted edge in S is j and this edge creates a lo
   S={d,j,k,m}
---------------------------------------------------------------------------
WHILE (11) (|T|=7<8 and S={∅})
   the least weighted edge in S is e and this edge creates a lo
   S={d,j,k,m}
---------------------------------------------------------------------------
WHILE (11) (|T|=7<8 and S={∅})
   the least weighted edge in S is d and this edge creates a loop
   S={j,k,m}
---------------------------------------------------------------------------
WHILE (12) (|T|=7<8 and S={∅})
   the least weighted edge in S is m and this edge creates a loop
   S={j,k}
---------------------------------------------------------------------------
WHILE (13) (|T|=7<8 and S={∅})
   the least weighted edge in S is k and this edge does not create a loop
   T={b,h,i,a,c,n,g,k}, S={k}
---------------------------------------------------------------------------
```

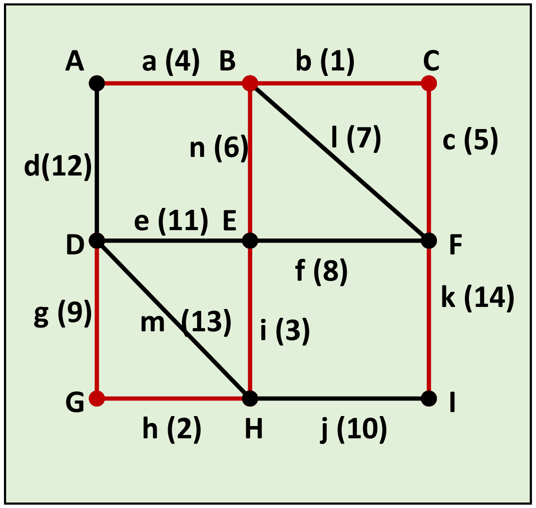


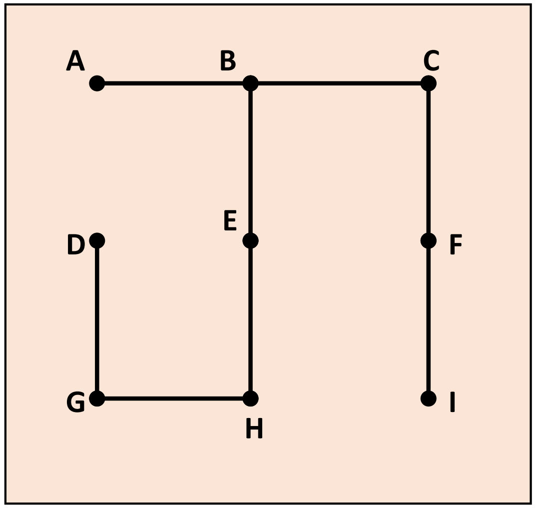
Spanning Tree…

➢ Kruskal Algorithm…

➢ Solution:

```
   ----------------------------------------------------------------------------
  WHILE (9) (|T|=8)
   END WHILE
   ----------------------------------------------------------------------------
STEP 3 T={b,h,i,a,c,n,g,k} is the minimum spanning tree and total weight is 38
```





References

- K.H. Rosen, Discrete Mathematics and Its Applications, Seventh Edition, Mc Graw Hill, 2012.
- R.P. Grimaldi, Discrete and Combinatorial Mathematcics, An Applied Introduction, Fifth Edition, Pearson, 2003.
- S.S. Epp, Discrete Mathematics with Applications, Fouth Edition, 2010.
- N. Yurtay, "Ayrık İşlemsel Yapılar" Lecture Notes, Sakarya University.