Activity 1. 0/1 Knapsack problem with constraints

An airline cargo company has an airplane that flies from the UK to Spain on a daily basis to transport some cargo. Before the flight, it receives bids for deliveries from (many) customers. Each bid contains the weight of the cargo item to be delivered, the amount costumers are willing to pay and some other observations. The airline is *constrained* by the total amount of weight the plane is allowed to carry. The company must choose a subset of the packages (bids) to carry on the plane in order to **maximize the total profit**, taking into account the **weight limit** that they must respect and all **other constraints**.

Bid	Weight (tons)	total price	Observations
1	2	200 €	This bid is compulsory.
2	15	500 €	Bids 2 and 9 cannot be selected together.
3	15	600 €	Bid 3 can only be chosen if both bids 2 and 6 are selected together.
4	42	1000 €	Bids 4 and 5 must be chosen together, or both or none.
5	15	300 €	
6	10	600 €	It is compulsory to choose exactly two bids out of these three bids.
7	20	800 €	
8	25	800 €	
9	5	300 €	Bid 9 can only be chosen if bid 8 is already selected.
10	16	600 €	
11	15	600 €	It has to choose at least one of these two bids.
12	23	1000 €	

If the airplane capacity is $W = 100$ tons, which bids have to be chosen in order to maximize the profit?

Activity 1   0/1 Knapsack problem with constraints

```
% example 0/1 knapsack
w=[5 4 3 4];          % weights
v=[6 5 3 5];          % values
W=13;                 % capacity

A=w;
b=W;
lb=zeros(1,4);ub=ones(1,4);
[sol, val]=intlinprog(-v,1:4,A,b,[],[],lb,ub)
```

$$x_i \in \{0,1\}$$

$$\sum_{j=1}^{n} v_j \cdot x_j$$

$$s.t. \sum_{j}^{n} w_j \cdot x_j \leq W$$

```
Optimization terminated.

sol =

     1
     1
     0
     1


val =

   -16
```

Additional constraint: select at most one of {1,2}

```
% example 0/1 knapsack with constraint
w=[5 4 3 4];          % weights
v=[6 5 3 5];          % values
W=13;                 % capacity

A=[w;
  1 1 0 0];
b=[W 1];
lb=zeros(1,4);ub=ones(1,4);
[sol, val]=intlinprog(-v,1:4,A,b,[],[],lb,ub)
```

$$x_i \in \{0,1\}$$

$$\sum_{j=1}^{n} v_j \cdot x_j$$

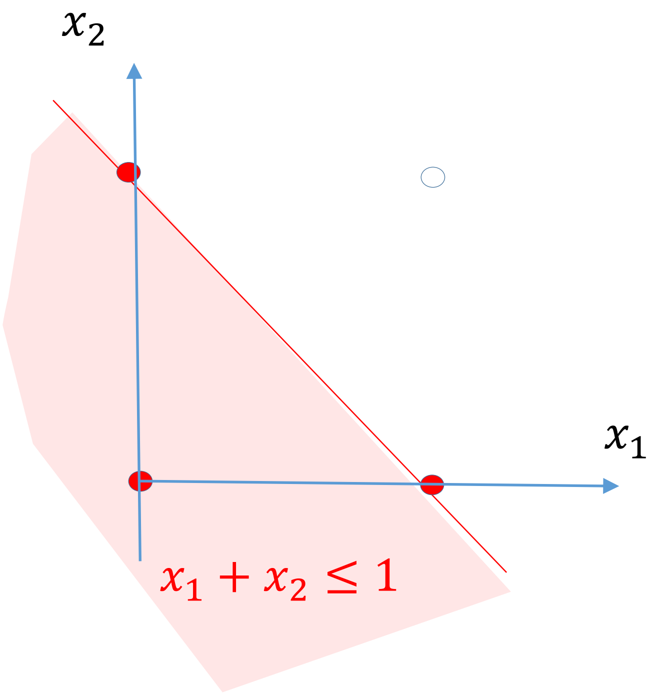
$$s.t. \sum_{j}^{n} w_j \cdot x_j \leq W$$

$$\boxed{x_1 + x_2 \leq 1}$$

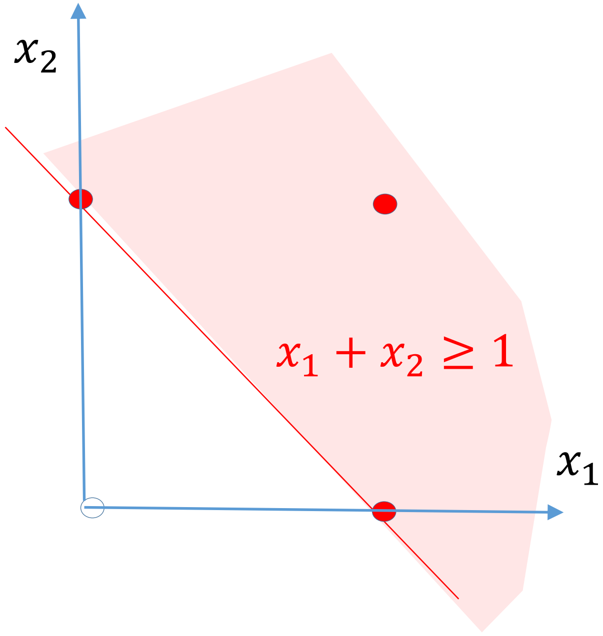
```
Optimization terminated.

sol =

     1
     0
     1
     1


val =

   -14
```

Activity 1 0/1 Knapsack problem with constraints

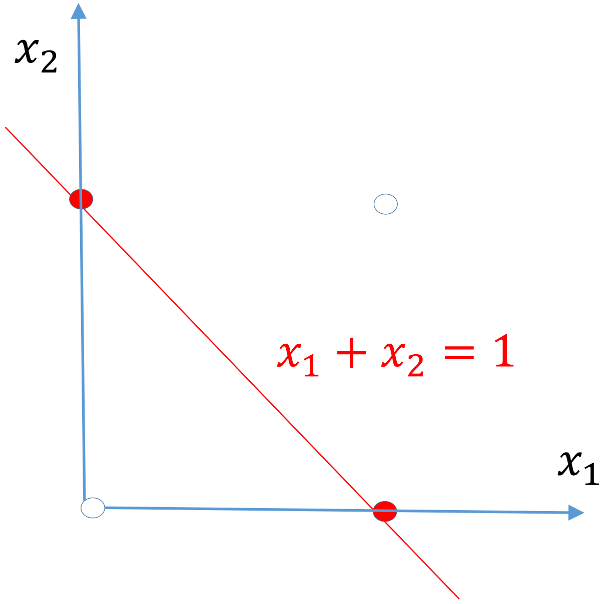
select **at most one** of {1 2}



select **at least one** of {1 2}

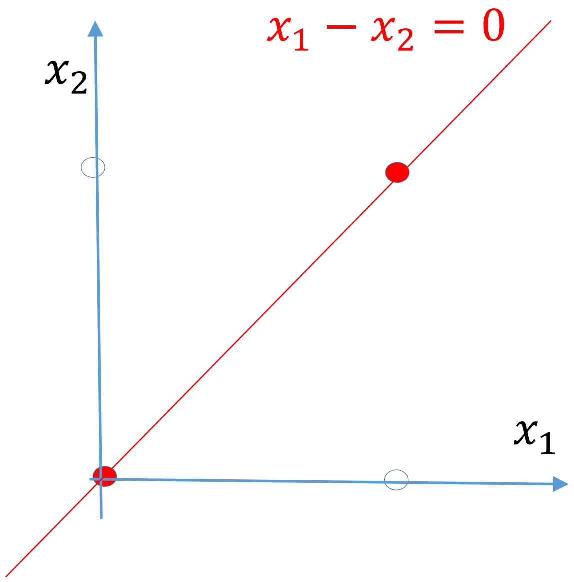


select **exactly one** of {1 2}

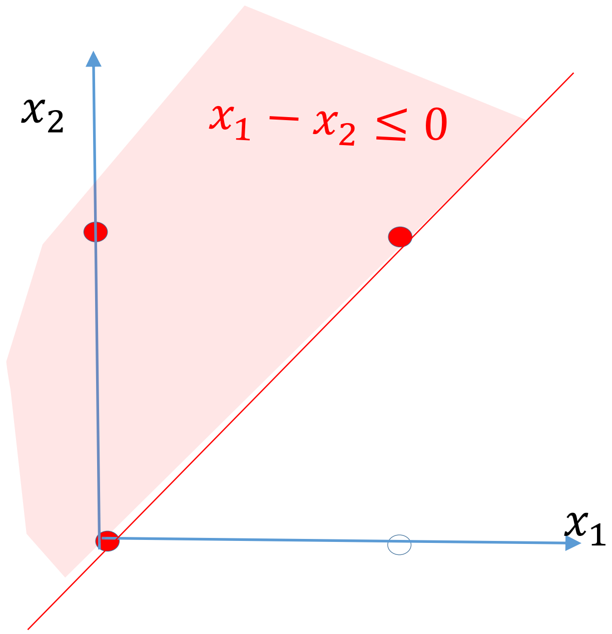


Activity 1   0/1 Knapsack problem with constraints

select **both or none** of {1 2}

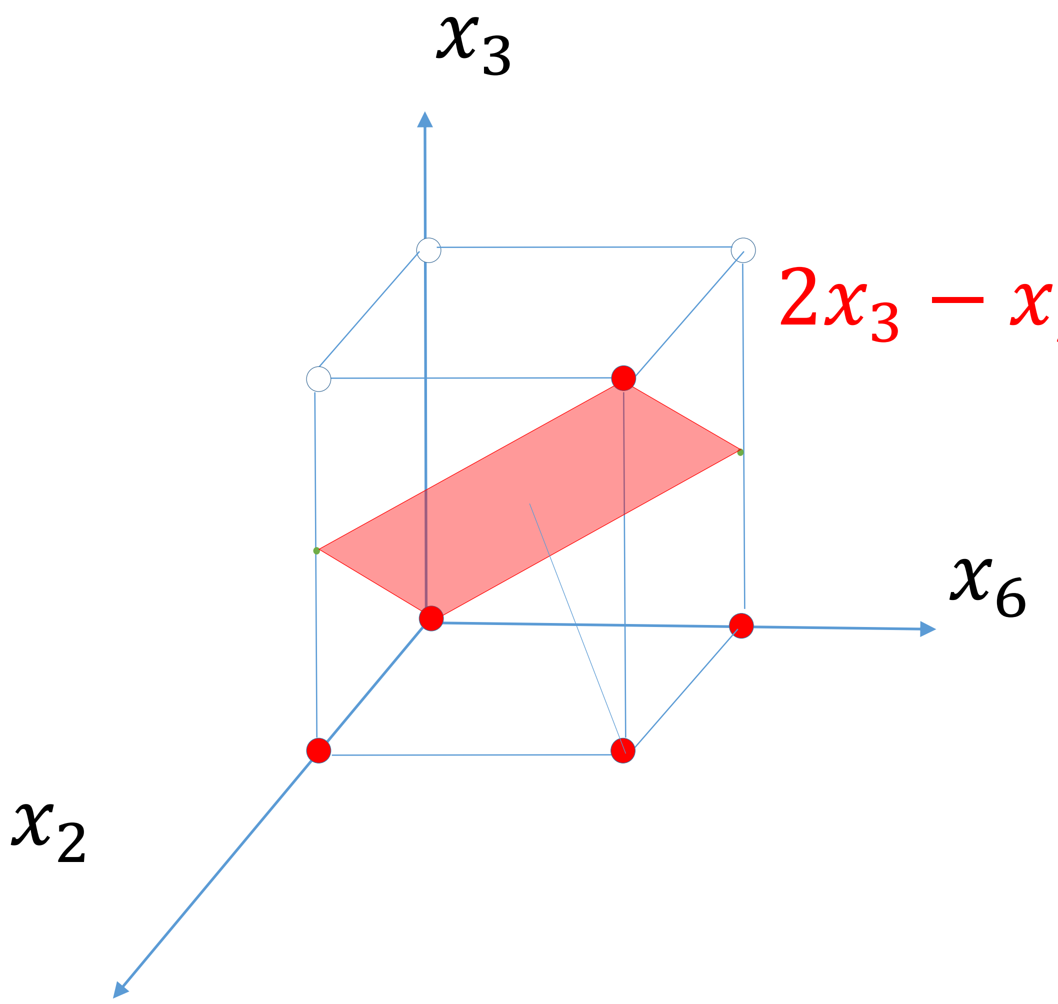


1 can be selected **only if** 2 is selected



Activity 1   0/1 Knapsack problem with constraints

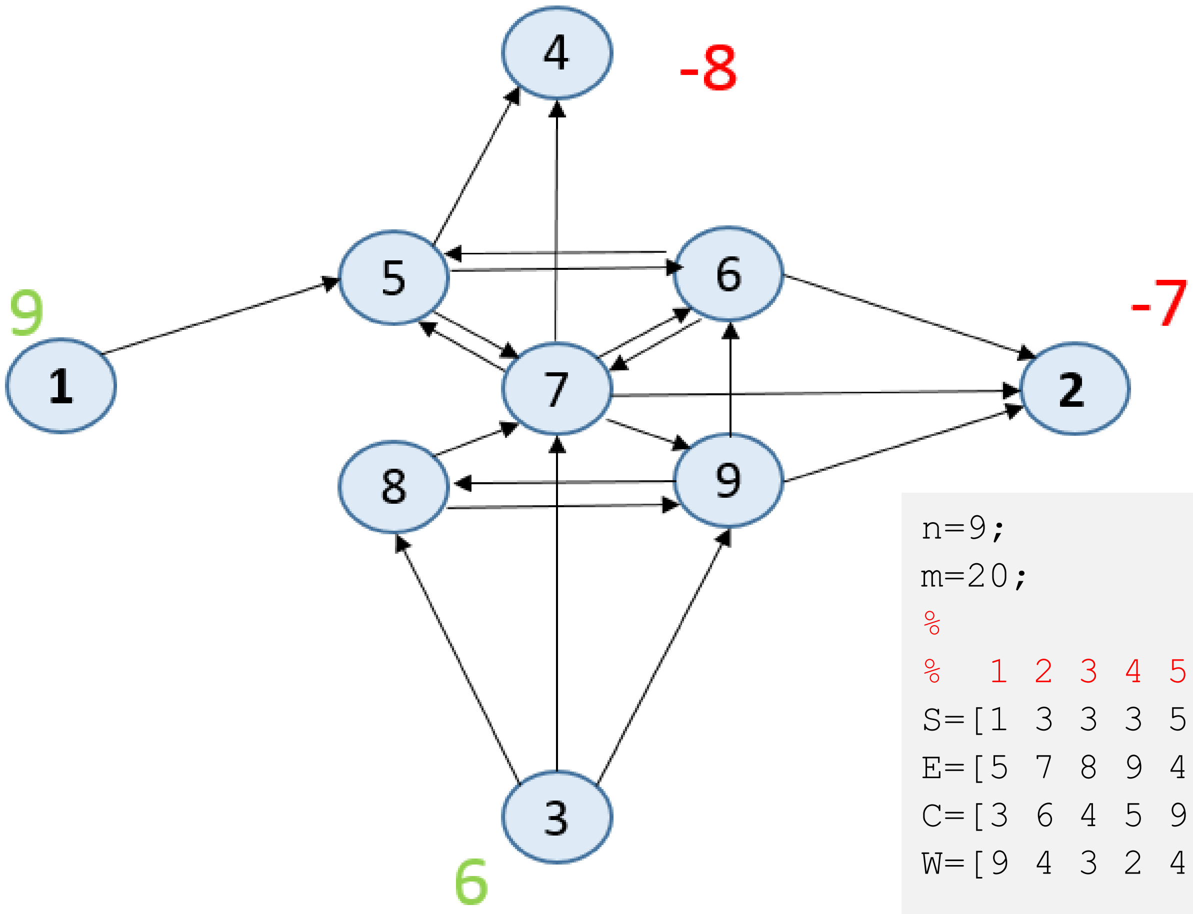
Bid 3 can **only** be chosen **if both** bids 2 and 6 are selected together.



$$2x_3 - x_2 - x_6 \leq 0$$

Activity 2. Minimum cost flow

Solve the following instance of the **minimum cost flow**, that is, calculate the flow at each of the 20 directed edges to supply goods from supply nodes 1 and 3 to demand nodes 2 and 4 satisfying the capacity constraints and that minimize the total cost.

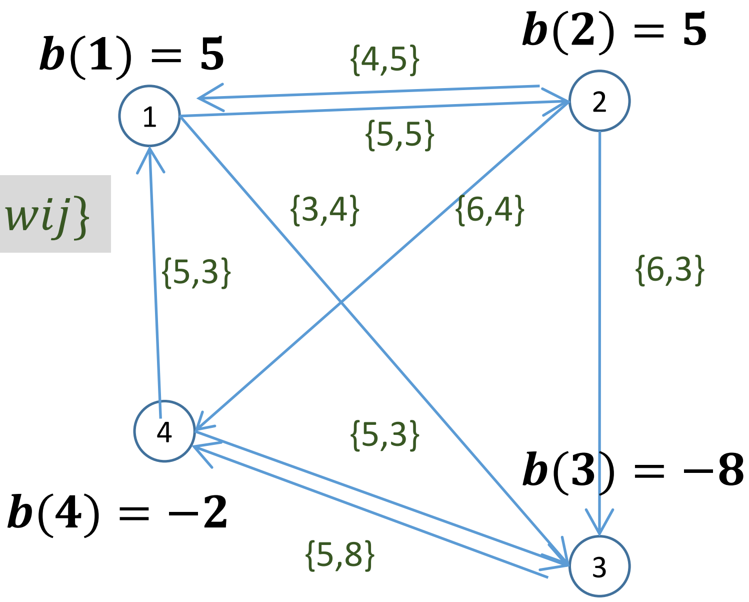



```
n=9;                                  % # nodes
m=20;                                 % # edges
%                     1 1 1 1 1 1 1 1 1 1 1
%  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0   edge
S=[1 3 3 3 5 5 5 6 6 6 7 7 7 7 7 8 8 9 9 9]; % starting node
E=[5 7 8 9 4 6 7 2 5 7 2 4 5 6 9 7 9 2 6 8]; % ending node
C=[3 6 4 5 9 5 4 6 6 4 5 4 7 3 5 5 3 5 2 5]; % cost
W=[9 4 3 2 4 2 5 4 5 4 5 4 5 2 4 5 4 3 5 3]; % capacity

%  1  2 3  4 5 6 7 8 9                        node
B=[9 -7 6 -8 0 0 0 0 0 ];                  % supplies and demands
```

Activity 2 Minimum cost flow

Let $G = (N, E)$ be a directed network
$b(i)$= supply or demand of node i (>0 supply, <0 demand)
Balance problem $\Rightarrow$ sum of $b(i)$ for all i equal 0.
c_{ij} cost of unit flow by edge ij
w_{ij} capacity of the edge ij
We seek a flow with the minimum cost satisfying the capacity constraints.

$\{c_{ij}, wij\}$



edge	c	w	
12	5	5	
13	3	4	
21	4	5	
23	6	3	
24	6	4	
34	5	8	
41	5	3	
43	5	3	

x_{ij}=flow by the edge ij

$$\min \sum_{i,j} c_{ij} \cdot x_{ij}$$

$$\text{st} \sum_{j:(kj\in E)} x_{kj} - \sum_{j:(jk\in E)} x_{ik} = b(k) \quad k \in n$$

$$0 \leq x_{ij} \leq w_{ij}$$

x_{ij} positive integers

Activity 2 Minimum cost flow

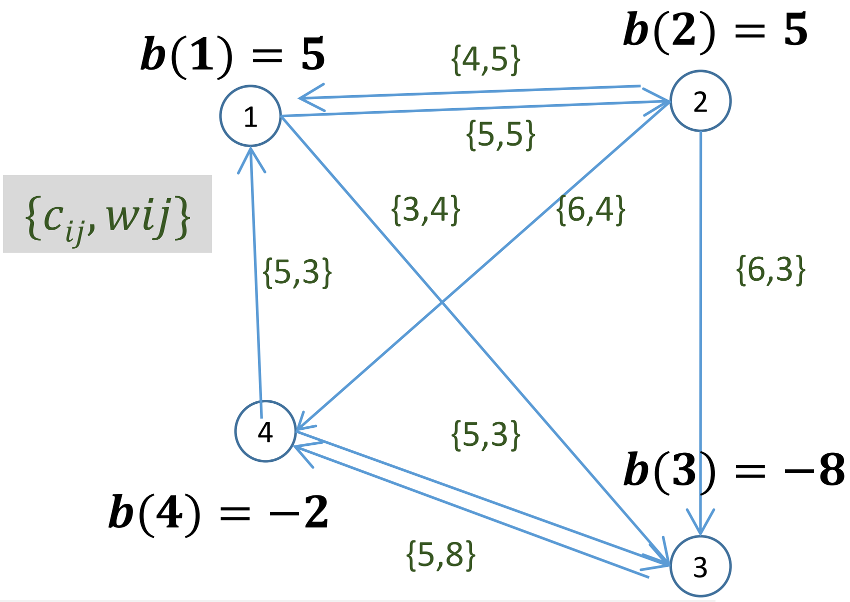
x_{ij}=flow by the edge ij

$$\min \sum_{i,j} c_{ij} \cdot x_{ij}$$

$$\text{st} \sum_{j:(kj \in E)} x_{kj} - \sum_{j:(jk \in E)} x_{ik} = b(k) \quad k \in n$$

$$0 \leq x_{ij} \leq w_{ij}$$

x_{ij} positive integers



edge	c	w	
12	5	5	
13	3	4	
21	4	5	
23	6	3	
24	6	4	
34	5	8	
41	5	3	
43	5	3	

```
%       1   1   2   2   2   3   4   4
%       2   3   1   3   4   4   1   3
C= [    5   3   4   6   6   5   5   5]';      % cost of unit of flow
W= [    5   4   5   3   4   8   3   3]';      % capacity

B= [    5   5  -8   -2];                      % supplies and demands
```

Activity 2 Minimum cost flow

x_{ij}=flow by the edge ij

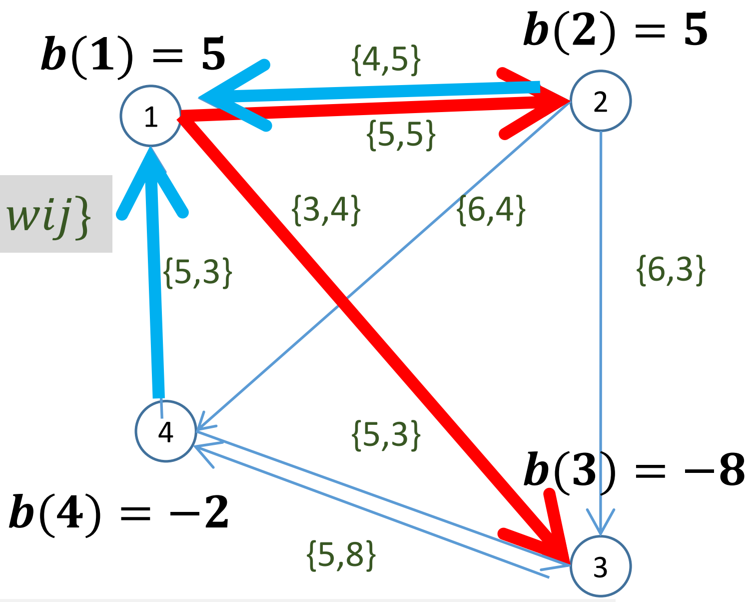
$$\min \sum_{i,j} c_{ij} \cdot x_{ij}$$

$$\boxed{\text{st} \sum_{j:(kj\in E)} x_{kj}} - \boxed{\sum_{j:(jk\in E)} x_{ik}} = b(k) \quad k \in n$$

$$0 \le x_{ij} \le w_{ij}$$

x_{ij} positive integers

$\{c_{ij}, wij\}$



edge	c	w	
12	5	5	
13	3	4	
21	4	5	
23	6	3	
24	6	4	
34	5	8	
41	5	3	
43	5	3	

```
%     1   1   2   2   2   3   4   4          % starting node
%     2   3   1   3   4   4   1   3          % ending node
C= [  5   3   4   6   6   5   5   5]';       % cost of unit of flow
W= [  5   4   5   3   4   8   3   3]';       % capacity

B= [  5   5   -8   -2];                      % supplies and demands
Aeq= [ 1   1  -1   0   0   0  -1   0;        % constraint for node 1
      -1   0   1   1   1   0   0   0;        % constraint for node 2
       0  -1   0  -1   0   1   0  -1;        % constraint for node 3
       0   0   0   0  -1  -1   1   1];       % constraint for node 4
beq=B;                                       % RHS of constraints
```

Activity 2 Minimum cost flow

x_{ij}=flow by the edge ij

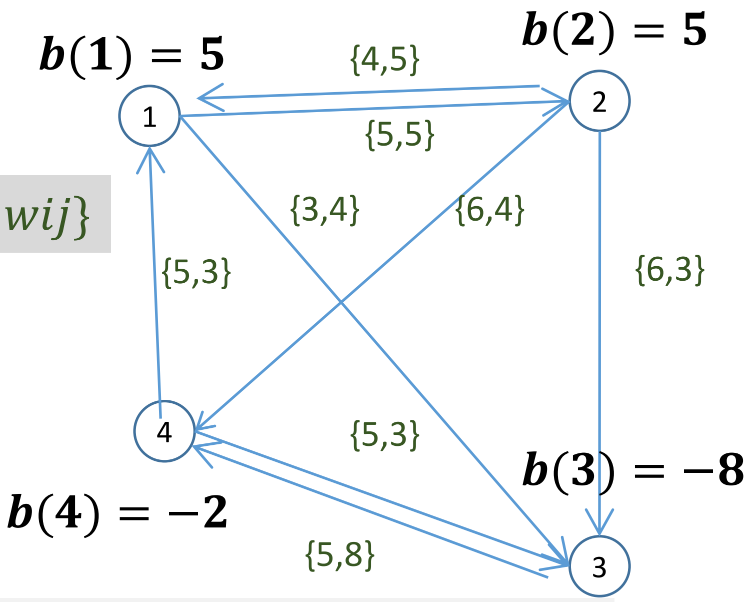
$$\min \sum_{i,j} c_{ij} \cdot x_{ij}$$

$$\text{st} \sum_{j:(kj\in E)} x_{kj} - \sum_{j:(jk\in E)} x_{ik} = b(k) \quad k \in n$$

$$\boxed{0 \le x_{ij} \le w_{ij}}$$

x_{ij} positive integers

$\{c_{ij}, wij\}$



edge	c	w	
12	5	5	
13	3	4	
21	4	5	
23	6	3	
24	6	4	
34	5	8	
41	5	3	
43	5	3	

```
%      1   1   2   2   2   3   4   4         % starting node
%      2   3   1   3   4   4   1   3         % ending node
C= [   5   3   4   6   6   5   5   5]';      % cost of unit of flow
W= [   5   4   5   3   4   8   3   3]';      % capacity

B= [   5   5  -8  -2];                       % supplies and demands
Aeq=[  1   1  -1   0   0   0  -1   0;        % constraint for node 1
      -1   0   1   1   1   0   0   0;        % constraint for node 2
       0  -1   0  -1   0   1   0  -1;        % constraint for node 3
       0   0   0   0  -1  -1   1   1];       % constraint for node 4
beq=B;                                       % RHS of constraints
lb = [ 0 0 0 0 0 0 0 0]';                   % variables must >=0
ub=W;                                        % ub=capacity
```

Activity 2 Minimum cost flow

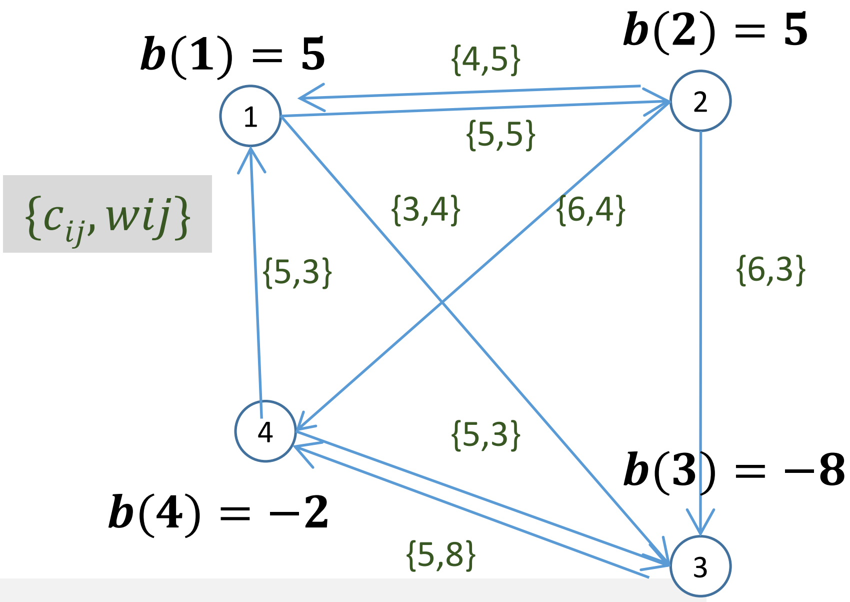
x_{ij}=flow by the edge ij

$$\min \sum_{i,j} c_{ij} \cdot x_{ij}$$

$$\text{st} \sum_{j:(kj \in E)} x_{kj} - \sum_{j:(jk \in E)} x_{ik} = b(k) \quad k \in n$$

$$0 \leq x_{ij} \leq w_{ij}$$

x_{ij} positive integers



edge	c	w	
12	5	5	1
13	3	4	4
21	4	5	0
23	6	3	3
24	6	4	3
34	5	8	0
41	5	3	0
43	5	3	1

```
%      1   1   2   2   2   3   4   4         % starting node
%      2   3   1   3   4   4   1   3         % ending node
C= [   5   3   4   6   6   5   5   5]';      % cost of unit of flow
W= [   5   4   5   3   4   8   3   3]';      % capacity

B= [   5   5  -8   -2];                      % supplies and demands
Aeq=[ 1   1 -1   0   0   0 -1   0;           % constraint for node 1
     -1   0   1   1   1   0   0   0;         % constraint for node 2
      0 -1   0 -1   0   1   0 -1;            % constraint for node 3
      0   0   0   0 -1 -1   1   1];          % constraint for node 4
beq=B;                                       % RHS of constraints
lb = [ 0 0 0 0 0 0 0 0]';                   % variables must >=0
ub=W;                                        % ub=capacity
[xmin, fval, flag]=linprog (C,[],[],Aeq,beq,lb,ub)
```

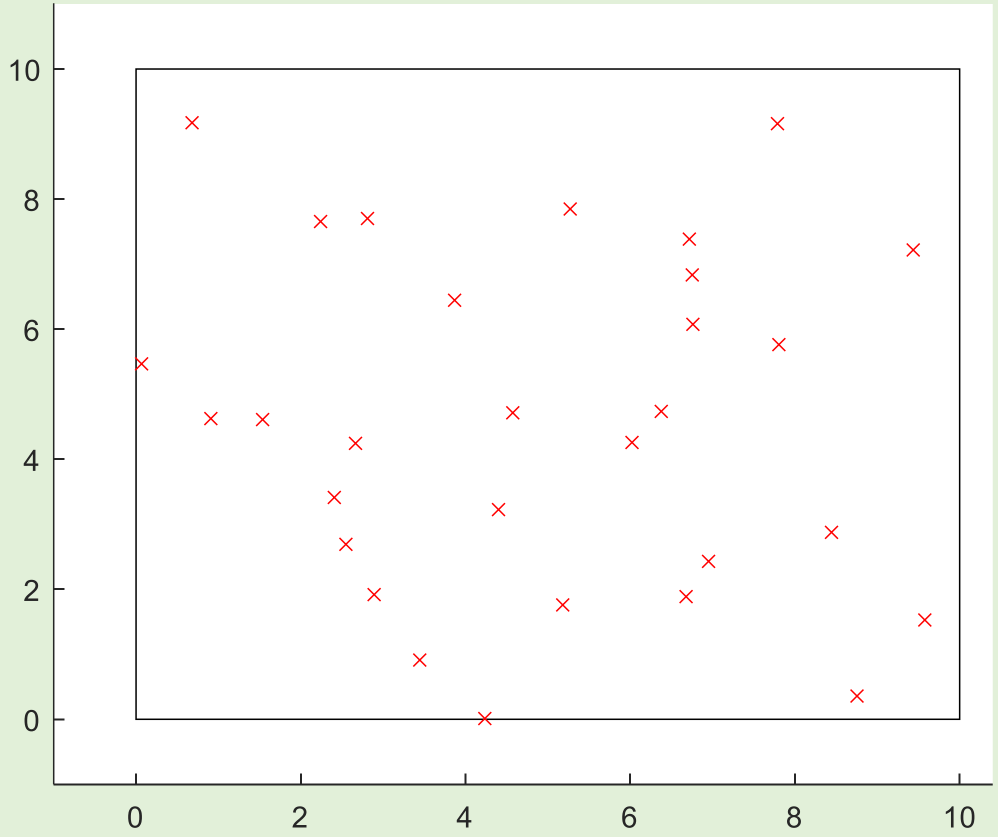
```
xmin =

   1
   4
   0
   3
   3
   0
   0
   1

fmin =
   58
```

Activity 3 Traveling salesperson problem (TSP)

Use binary integer programming to solve the classic **traveling salesman problem**. This problem involves finding the shortest closed tour through a set of points.

a) Generate a set of 30 random points inside the square of opposite vertices (0,0) and (10,10).

```
n=30;                           % number of points
P=zeros(n,2);                   % inicialize point's coordinates
D=zeros(n,n);                   % inicialize distance matrix


P=rand(n,2)*10;                  % generate n random points

rectangle('Position',[0 0 10 10])
axis([-1 11 -1 11])
hold on
plot(P(:,1),P(:,2),'rx')        % draw the points as red crosses

for i=1:(n-1)                   % calculate distance matrix
    for j=(i+1):n
        D(i,j)=norm(P(i,:)-P(j,:));
        D(j,i)=D(i,j);
    end
end
```



Activity 3 Traveling salesperson problem (TSP)

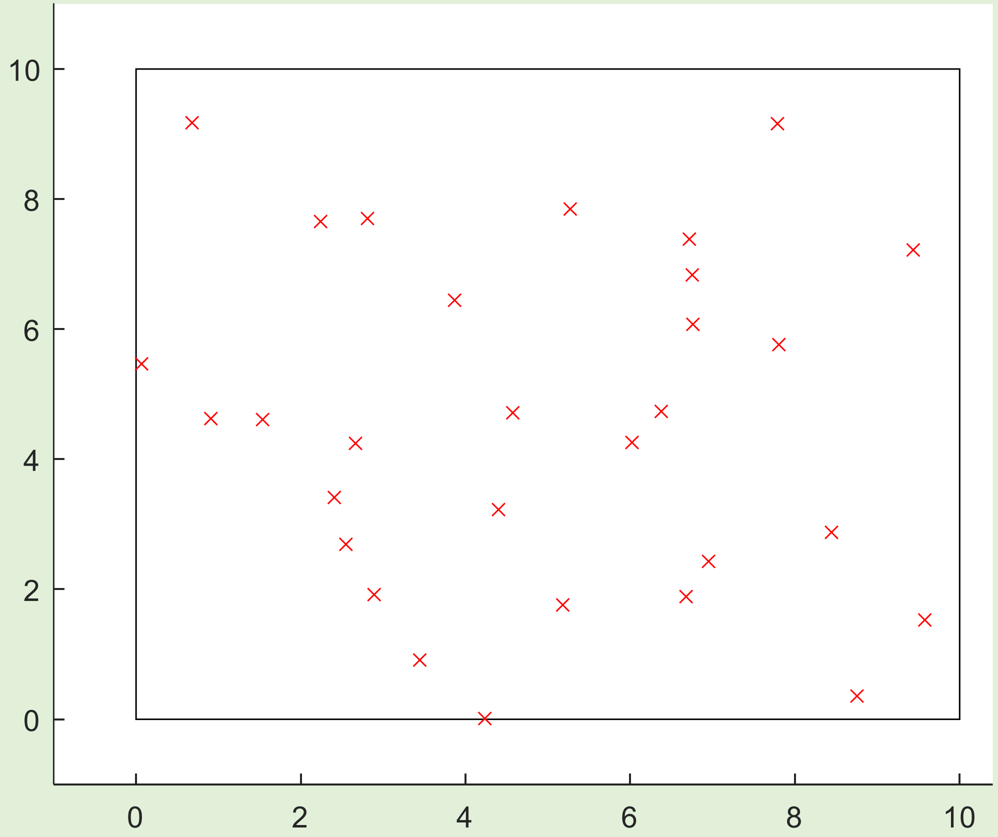
Use binary integer programming to solve the classic **traveling salesman problem**. This problem involves finding the shortest closed tour through a set of points.

b) Considering n^2 binary variables x_{ij}, solve the following binary linear problem:

$$\min \sum_{i=1}^{n} d_{ij} \cdot x_{ij}$$

$$\text{st} \sum_{j}^{n} x_{ij} = 1 \qquad i = 1 \dots n$$

$$\text{st} \sum_{i}^{n} x_{ij} = 1 \qquad j = 1 \dots n$$



c) Add suitable integer variables and constraints to the previous problem to found the shortest closed tour passing through all points.

Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1:m$

$x_i = 1$ if directed edge i is selected

values: $\{v_1, \ldots, v_m\}$ length of directed edges.

For each node i

Sum of edges starting at i equal to 1

Sum of edges finishing at i equal to 1

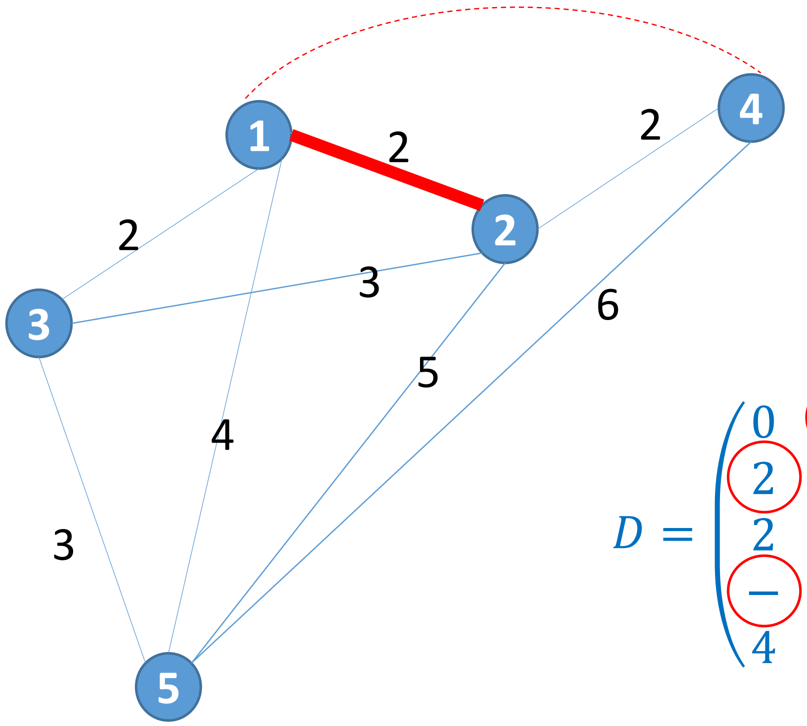
$$\min \sum_{i=1}^{m} v_i \cdot x_i$$

$$\text{st} \quad \sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \ldots n$$

$$\sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \ldots n$$

$$x_i \in \{0,1\}$$

```
% Traveling salesperson problem
n=5;m=16;
S=[ 1 1 1 2 2 2 2 3 3 3 4 4 5 5 5 5]; % starting node
E=[ 2 3 5 1 3 4 5 1 2 5 2 5 1 2 3 4]; % ending node
D=[ 2 2 4 2 3 2 5 2 3 3 2 6 4 5 3 6]; % distance
```



$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$

Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1{:}m$
$x_i = 1$ if <u>directed edge</u> i is selected
values: $\{v_1, \ldots, v_m\}$ length of directed edges.

For each node i
Sum of edges starting at i equal to 1
Sum of edges finishing at i equal to 1

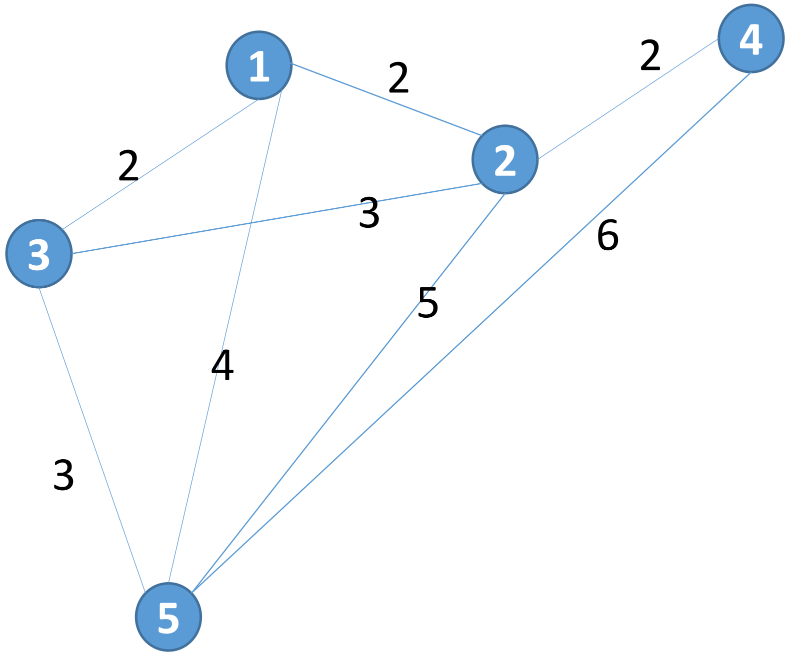
$$\min \sum_{i=1}^{m} v_i \cdot x_i$$

$$\boxed{\sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \ldots n}$$

$$\text{st} \sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \ldots n$$

$$x_i \in \{0,1\}$$

$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$




```
% Traveling salesperson problem
n=5;m=16;
S=[ 1  1  1  2  2  2  2  3  3  3  4  4  5  5  5  5]; % starting node
E=[ 2  3  5  1  3  4  5  1  2  5  2  5  1  2  3  4]; % ending node
D=[ 2  2  4  2  3  2  5  2  3  3  2  6  4  5  3  6]; % distance
```

```
A1=[1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0;
    0  0  0  1  1  1  1  0  0  0  0  0  0  0  0  0;
    0  0  0  0  0  0  0  1  1  1  0  0  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1];

A2=[0  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0;
    1  0  0  0  0  0  0  0  1  0  1  0  0  1  0  0;
    0  1  0  0  1  0  0  0  0  0  0  0  0  0  1  0;
    0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1;
    0  0  1  0  0  0  1  0  0  1  0  1  0  0  0  0];
Aeq=[A1;A2];
beq=ones(2*n,1);
```

Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1:m$
$x_i = 1$ if <u>directed edge</u> i is selected
values: $\{v_1, \dots, v_m\}$ length of directed edges.

For each node i
Sum of edges starting at i equal to 1
Sum of edges finishing at i equal to 1

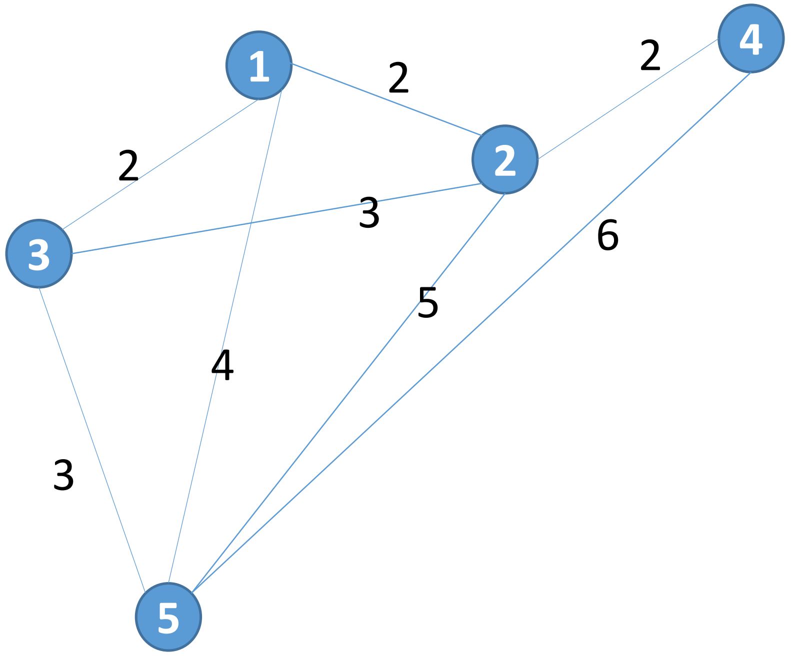
$$\min \sum_{i=1}^{m} v_i \cdot x_i$$

$$\text{st} \quad \sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \dots n$$

$$\boxed{\sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \dots n}$$

$$x_i \in \{0,1\}$$

$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$




```
% Traveling salesperson problem
n=5;m=16;
S=[ 1  1  1  2  2  2  2  3  3  3  4  4  5  5  5  5]; % starting node
E=[ 2  3  5  1  3  4  5  1  2  5  2  5  1  2  3  4]; % ending node
D=[ 2  2  4  2  3  2  5  2  3  3  2  6  4  5  3  6]; % distance
```

```
A1=[1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0;
    0  0  0  1  1  1  1  0  0  0  0  0  0  0  0  0;
    0  0  0  0  0  0  0  1  1  1  0  0  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1];

A2=[0  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0;
    1  0  0  0  0  0  0  0  1  0  1  0  0  1  0  0;
    0  1  0  0  1  0  0  0  0  0  0  0  0  0  1  0;
    0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1;
    0  0  1  0  0  0  1  0  0  1  0  1  0  0  0  0];
Aeq=[A1;A2];
beq=ones(2*n,1);
```

Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1{:}m$
$x_i = 1$ if <u>directed edge</u> i is selected
values: $\{v_1, \dots, v_m\}$ length of directed edges.

For each node i
Sum of edges starting at i equal to 1
Sum of edges finishing at i equal to 1

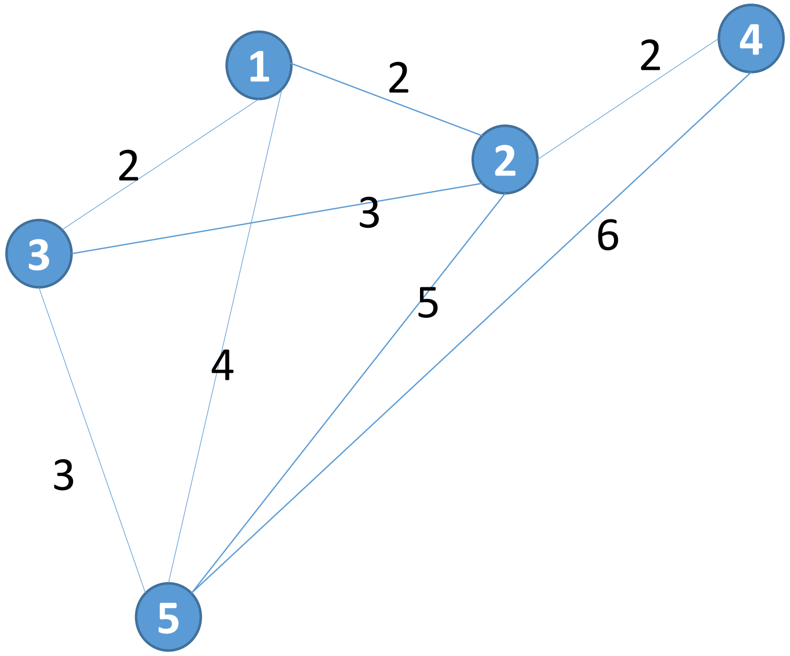
$$\min \sum_{i=1}^{m} v_i \cdot x_i$$

$$\sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \dots n$$

$$\text{st} \quad \sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \dots n$$

$$\boxed{x_i \in \{0,1\}}$$

$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$




```
% Traveling salesperson problem
n=5;m=16;
S=[ 1  1  1  2  2  2  2  3  3  3  4  4  5  5  5  5]; % starting node
E=[ 2  3  5  1  3  4  5  1  2  5  2  5  1  2  3  4]; % ending node
D=[ 2  2  4  2  3  2  5  2  3  3  2  6  4  5  3  6]; % distance
```

```
A1=[1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0;
    0  0  0  1  1  1  1  0  0  0  0  0  0  0  0  0;
    0  0  0  0  0  0  0  1  1  1  0  0  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1];

A2=[0  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0;
    1  0  0  0  0  0  0  0  1  0  1  0  0  1  0  0;
    0  1  0  0  1  0  0  0  0  0  0  0  0  0  1  0;
    0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1;
    0  0  1  0  0  0  1  0  0  1  0  1  0  0  0  0];
Aeq=[A1;A2];
beq=ones(2*n,1);
lb=zeros(m,1);ub=ones(m,1);
```

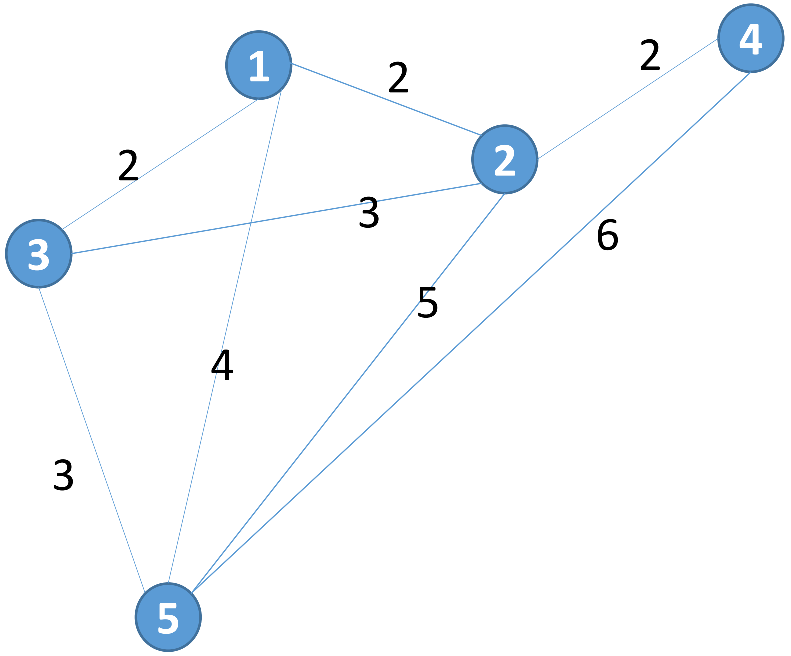
Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1{:}m$
$x_i = 1$ if <u>directed edge</u> i is selected
values: $\{v_1, \dots, v_m\}$ length of directed edges.

For each node i
Sum of edges starting at i equal to 1
Sum of edges finishing at i equal to 1

$$\min \sum_{i=1}^{m} v_i \cdot x_i$$
$$\sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \dots n$$
$$\text{st} \quad \sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \dots n$$
$$x_i \in \{0,1\}$$

$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$




```
% Traveling salesperson problem
n=5;m=16;
S=[ 1  1  1  2  2  2  2  3  3  3  4  4  5  5  5  5]; % starting node
E=[ 2  3  5  1  3  4  5  1  2  5  2  5  1  2  3  4]; % ending node
D=[ 2  2  4  2  3  2  5  2  3  3  2  6  4  5  3  6]; % distance
```

```
A1=[1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0;
    0  0  0  1  1  1  1  0  0  0  0  0  0  0  0  0;
    0  0  0  0  0  0  0  1  1  1  0  0  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0;
    0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1];

A2=[0  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0;
    1  0  0  0  0  0  0  0  1  0  1  0  0  1  0  0;
    0  1  0  0  1  0  0  0  0  0  0  0  0  0  1  0;
    0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1;
    0  0  1  0  0  0  1  0  0  1  0  1  0  0  0  0];
Aeq=[A1;A2];
beq=ones(2*n,1);
lb=zeros(m,1);ub=ones(m,1);
```

```
[x,fval]=intlinprog(D,1:m,[],[],Aeq,beq,lb,ub)
```

Activity 3 Traveling salesperson problem (TSP)

$x_i \in \{0,1\}$ binaries variables $\quad i = 1:m$
$x_i = 1$ if directed edge i is selected
values: $\{v_1, \dots, v_m\}$ length of directed edges.

For each node i
Sum of edges starting at i equal to 1
Sum of edges finishing at i equal to 1

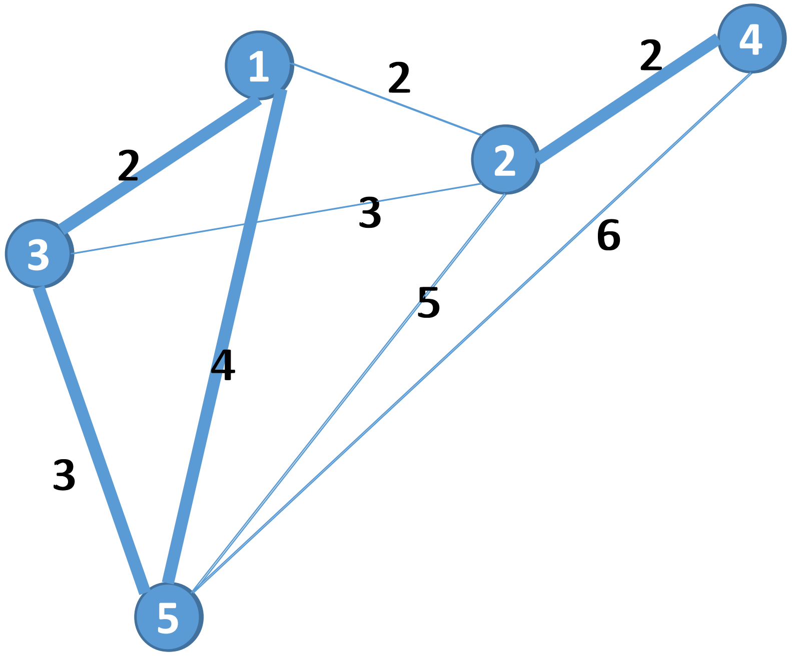
$$\min \sum_{i=1}^{m} v_i \cdot x_i$$

$$\sum_{j:(x_j \text{ starts at } i)} x_j = 1 \quad i = 1 \dots n$$

$$\text{st} \quad \sum_{j:(x_j \text{ ends at } i)} x_j = 1 \quad i = 1 \dots n$$

$$x_i \in \{0,1\}$$

$$D = \begin{pmatrix} 0 & 2 & 2 & - & 4 \\ 2 & 0 & 3 & 2 & 5 \\ 2 & 3 & 0 & - & 3 \\ - & 2 & - & 0 & 6 \\ 4 & 5 & 3 & 6 & 0 \end{pmatrix}$$




```
% Traveling salesperson problem
n=5;m=16;
S=[ 1 1 1 2 2 2 2 3 3 3 4 4 5 5 5 5]; % starting node
E=[ 2 3 5 1 3 4 5 1 2 5 2 5 1 2 3 4]; % ending node
D=[ 2 2 4 2 3 2 5 2 3 3 2 6 4 5 3 6]; % distance
```

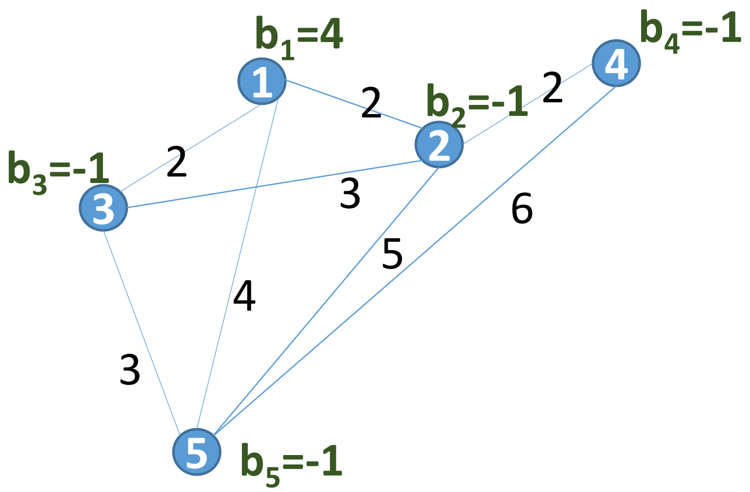
```
x= 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0
```

```
[x,fval]=intlinprog(D,1:m,[],[],Aeq,beq,lb,ub)
```

Activity 3 Traveling salesperson problem (TSP)

$x_j \in \{0,1\}$ binaries variables $j = 1:m$ (one variable per edge)
$x_j = 1$ if directed edge j is selected
values: $\{d_1, \ldots, d_m\}$ length of directed edges.

For each node k
Sum of edges starting at k equal to 1
Sum of edges finishing at k equal to 1



$n = \#$ *cities*

x_j boolean variables,
as many as edges (m)

y_j positive integer variables,
as many as edges (m)

$$\min \sum_{j=1}^{m} d_j \cdot x_j$$

$$\sum_{\substack{x_j \text{ starts} \\ \text{in city } k}} x_j = 1 \quad k = 1, \ldots, n$$

$$\sum_{\substack{x_j \text{ ends} \\ \text{in city } k}} x_j = 1 \quad k = 1, \ldots, n$$

$$st \sum_{\substack{x_j \text{ starts} \\ \text{in city } k}} y_j - \sum_{\substack{x_j \text{ ends} \\ \text{in city } k}} y_j = b(k) \quad k = 1, \ldots, n$$

$$0 \le y_k \le (n-1) \cdot x_k \quad j = 1, \ldots, m$$

$$x_j \in \{0, 1\} \quad j = 1, \ldots, m$$

$$y_j \in Z+ \quad j = 1, \ldots, m$$

Activity 3 Traveling salesperson problem (TSP)

$n = \#\ nodes$

x_j boolean variables, as many as edges (m)

y_j positive integer variables, as many as edges (m)

$$\min \sum_{j=1}^{m} d_j \cdot x_j$$

$$\sum_{\substack{x_j \text{ starts} \\ \text{in city } k}} x_j = 1 \qquad k = 1,\dots,n$$

$$\sum_{\substack{x_j \text{ ends} \\ \text{in city } k}} x_j = 1 \qquad k = 1,\dots,n$$

$$st \quad \sum_{\substack{x_j \text{ starts} \\ \text{in city } k}} y_j - \sum_{\substack{x_j \text{ ends} \\ \text{in city } k}} y_j = b(k) \qquad k = 1,\dots,n$$

$$y_j \le (n-1)\cdot x_j \qquad j = 1,\dots,m$$

$$x_j \in \{0\ ,1\} \qquad j = 1,\dots,m$$

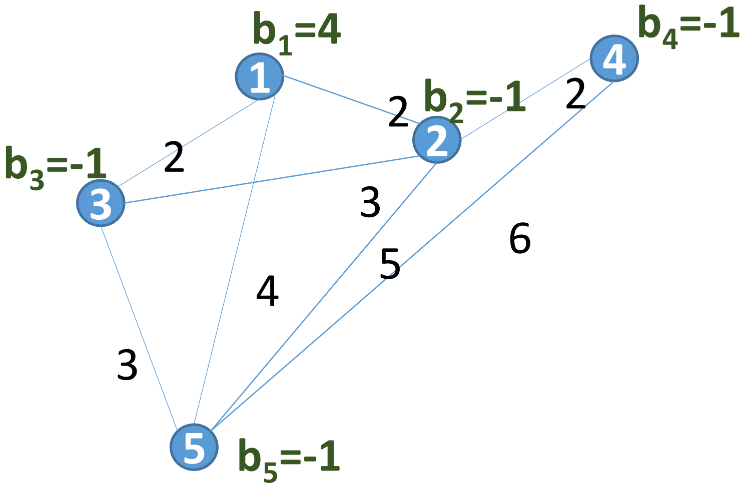
$$y_j \in Z+ \qquad j = 1,\dots,m$$

Aeq beq

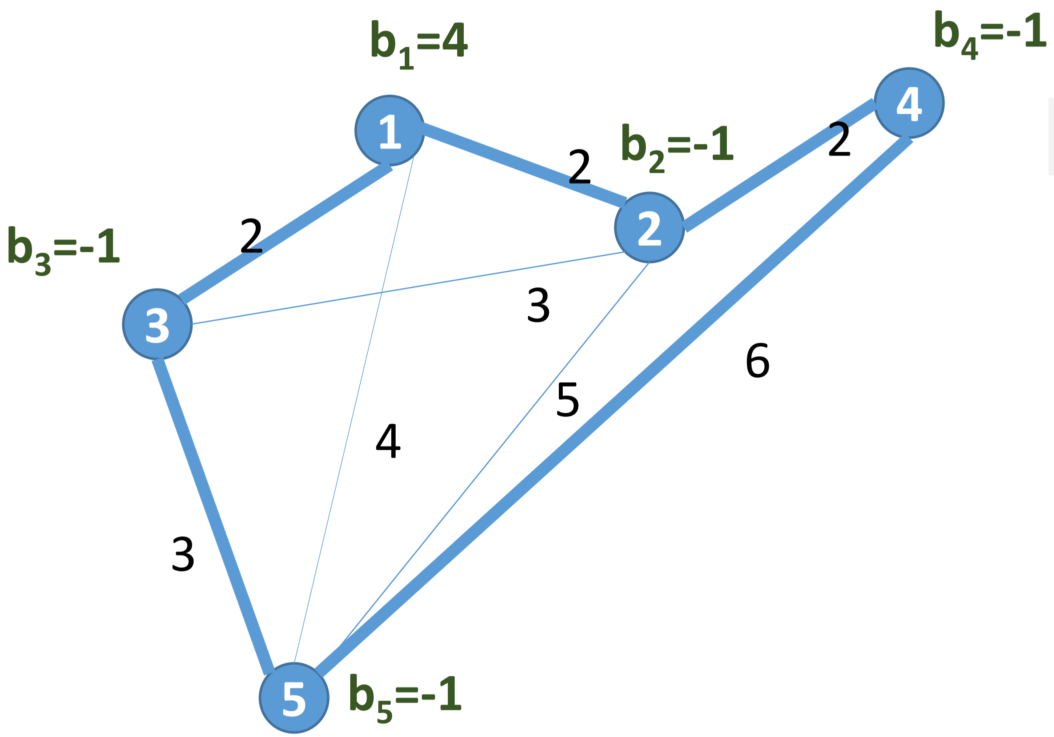
$$\left(\begin{array}{c|c} A_1 & \\ \hline A_2 & \\ \hline & A_3 \end{array}\right) = \begin{pmatrix} 1 \\ 1 \\ b(k) \end{pmatrix}$$

A b

$$\left(\begin{array}{c|c} \underbrace{-(n-1)\cdot I}_{x_1,\dots x_{n\cdot(n-1)}} & \underbrace{I}_{y_1,\dots y_{n\cdot(n-1)}} \end{array}\right) \le (0)$$



Activity 3 Traveling salesperson problem (TSP)




```
S=[ 1 1 1 2 2 2 2 3 3 3 4 4 5 5 5 5];
E=[ 2 3 5 1 3 4 5 1 2 5 2 5 1 2 3 4];
```

```
x =
0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1
0 4 0 0 0 0 0 0 0 3 1 0 0 0 0 2

fval=15
```

```
[x,fval]=intlinprog(f,1:(2*m),A,b,Aeq,beq,lb,ub)
```

Aeq beq

$$\left(\begin{array}{c|c} A_1 & \\ \hline A_2 & \\ \hline & A_3 \end{array}\right) = \begin{pmatrix} 1 \\ 1 \\ b(k) \end{pmatrix}$$

3n eq constr

A b

$$\left(\begin{array}{c|c} \underbrace{-(n-1)\cdot I}_{x_1,\ldots x_m} & \underbrace{I}_{y_1,\ldots y_m} \end{array}\right) \le (0)$$

m ineq constr

Activity 3 Traveling salesperson problem (TSP)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculates the vector f
%
% f= # of cities, D(i,j)=distance from city i to city j
% f must be a vector of 2·n·(n-1) coeficients associated to the variables
% x(1), x(2),...,x(n·(n-1)), y(1), y(2),...,y(n·(n-1))

f=reshape(D,1,n*n);     % writes the distance matrix D in a one dimensional vector
f=f(f~=0);              % deletes variables related with edges ii  (size(f)=n*(n-1))
f=[f,zeros(1,n*(n-1))];% add to the previous vector n·(n-1) coeficients associated to the y
variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Activity 3 Traveling salesperson problem (TSP)

```
% Calculate the matrix A and b
%        /  A1    |  Z    \             / 1 \
%  A= |          |         |       b= |      |
%     |   A2     |  Z      |          |  1   |
%     |          |         |          |      |
%     |   Z      |  A3     |          | B(I)|
%        \        |       /             \    /

%Aeq= ( A4    |  I4       )        beq=0
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the matrix A1
% example with n=4
% A1=(4  x  12) matrix
%      / 1 1 1 0 0 0 0 0 0 0 0 0 \
%      | 0 0 0 1 1 1 0 0 0 0 0 0 |
%   A1=| 0 0 0 0 0 0 1 1 1 0 0 0 |
%      \ 0 0 0 0 0 0 0 0 0 1 1 1 /
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the matrix A2
% example with n=4
% A2=(4  x  12) matrix
%      / 0 0 0 1 0 0 1 0 0 1 0 0 \
%      | 1 0 0 0 0 0 0 1 0 0 1 0 |
%   A2=| 0 1 0 0 1 0 0 0 0 0 0 1 |
%      \ 0 0 1 0 0 1 0 0 1 0 0 0 /
%
```

```
% A3=A1-A2;

% A4=-(n-1)*eye(n*(n-1));            %-(n-1)·I
```

Minimum spanning tree. Prim Algorithm

We want to select a set of scheduled services among these previous 20 airports so that we can go from any airport to any other airport using some of these services (probably making more than one making transfer through other airports).

A scheduled service consists of a flight from one airport to another. Always the return service exists, that means, for instance, if we can go from San Francisco to Los Angeles, there exist also the service from Los Angeles to San Francisco. That is equivalent to consider the graph as an undirected graph.

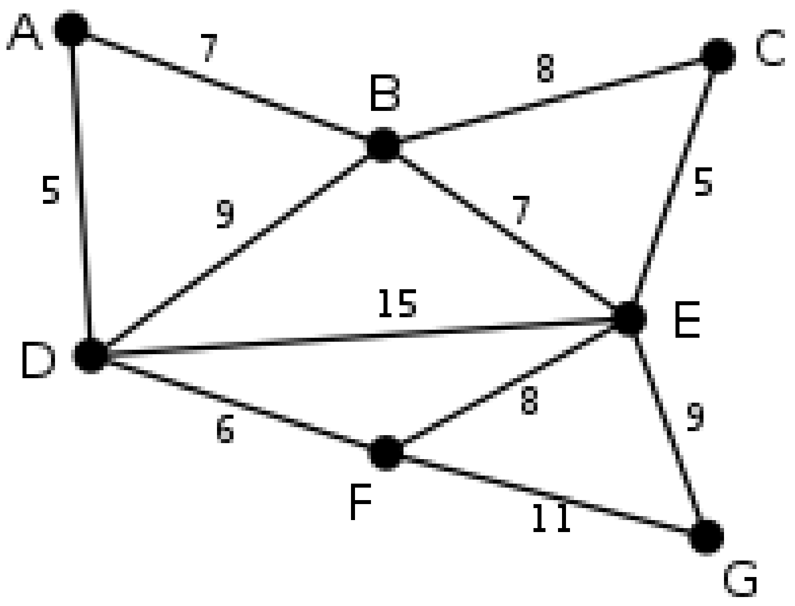
Which is the selection that minimize the sum of distances of all services?

Solve the problem by implementing the **Prim algorithm**.

```
n=20;
D=[0    144 114 105 31  109 135 132 85  79  158 20  73  162 127 190 156 58  87  71;
   144  0   144 181 147 76  195 73  64  114 220 135 71  18  39  60  37  101 62  146;
   114  144 0   49  86  169 51  78  130 42  76  94  114 154 105 151 125 137 94  46;
   105  181 49  0   73  189 31  124 152 67  52  88  135 195 146 197 169 147 123 40;
   31   147 86  73  0   128 104 119 97  57  126 17  82  164 122 184 151 80  85  40;
   109  76  169 189 128 0   212 126 38  128 238 112 54  92  95  137 110 51  77  148;
   135  195 51  31  104 212 0   129 174 85  26  118 157 206 157 201 176 173 141 67;
   132  73  78  124 119 126 129 0   92  65  153 115 84  80  35  73  47  118 55  98;
   85   64  130 152 97  38  174 92  0   90  200 82  17  82  66  120 89  36  39  112;
   79   114 42  67  57  128 85  65  90  0   111 59  73  128 80  137 106 95  57  33;
   158  220 76  52  126 238 26  153 200 111 0   141 183 231 182 224 201 198 167 91;
   20   135 94  88  17  112 118 115 82  59  141 0   67  153 114 177 142 63  75  52;
   73   71  114 135 82  54  157 84  17  73  183 67  0   90  64  123 89  35  28  95;
   162  18  154 195 164 92  206 80  82  128 231 153 90  0   49  47  35  119 79  161;
   127  39  105 146 122 95  157 35  66  80  182 114 64  49  0   62  28  99  40  113;
   190  60  151 197 184 137 201 73  120 137 224 177 123 47  62  0   34  156 102 170;
   156  37  125 169 151 110 176 47  89  106 201 142 89  35  28  34  0   123 68  139;
   58   101 137 147 80  51  173 118 36  95  198 63  35  119 99  156 123 0   63  106;
   87   62  94  123 85  77  141 55  39  57  167 75  28  79  40  102 68  63  0   85;
   71   146 46  40  40  148 67  98  112 33  91  52  95  161 113 170 139 106 85  0];
```

Minimum spanning tree. Prim Algorithm

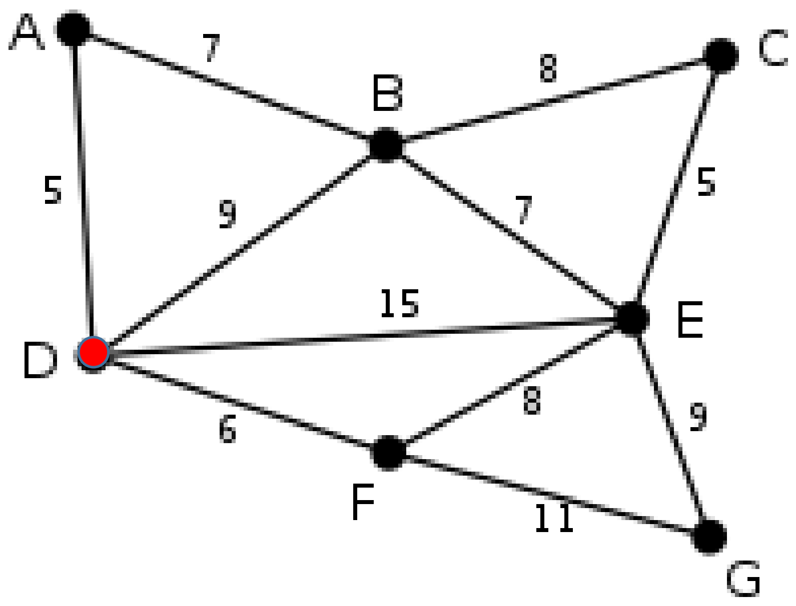
Prim's algorithm, was originally discovered in 1930 by mathematician Vojtěch Jarník and later independently by Prim in 1957. It was later rediscovered by Edsger Dijkstra in 1959. It is sometimes referred to as the *DJP algorithm*.



The algorithm increases the size of a tree, one vertex at a time, starting with a tree consisting of a single vertex, until it spans all vertices.

Initialize: V_{new} = {x}, where x is an arbitrary node (starting point) from V, E_{new} = {}
Repeat until $V_{new} = V$:
Choose an edge {u, v} with minimal weight such that u is in V_{new} and v is not
Add v to V_{new}, and {u, v} to E_{new}
Output: E_{new} describe a minimal spanning tree

Example: Minimum spanning tree. Prim Algorithm



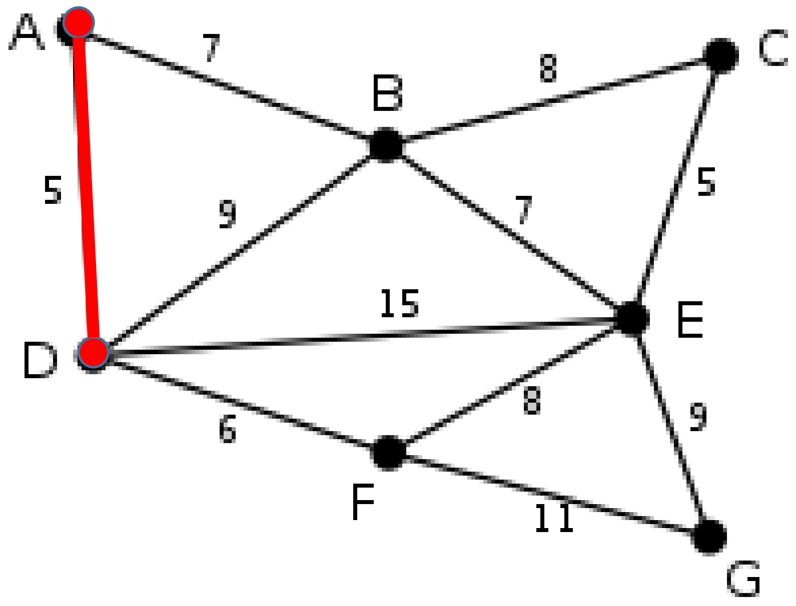
V_{old} {A,B,C,E,F,G}

V_{new} {D}

DA=5
DB=9
DE=15
DF=6

$$E_{new} = \{DA\}$$

Example: Minimum spanning tree. Prim Algorithm



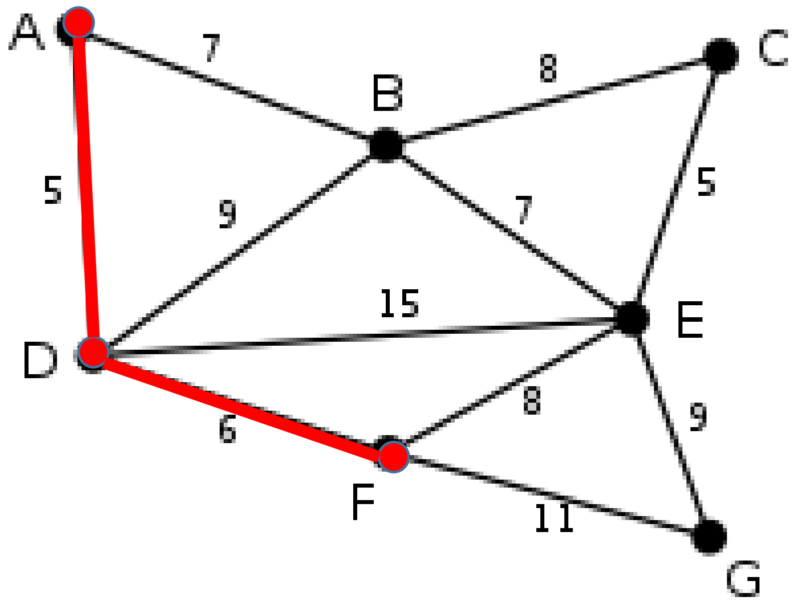
{B,C,E,F,G}

{D,A}

AB=7
DB=9
DE=15
DF=6

$$E_{new} = \{DA, DF\}$$

Example: Minimum spanning tree. Prim Algorithm



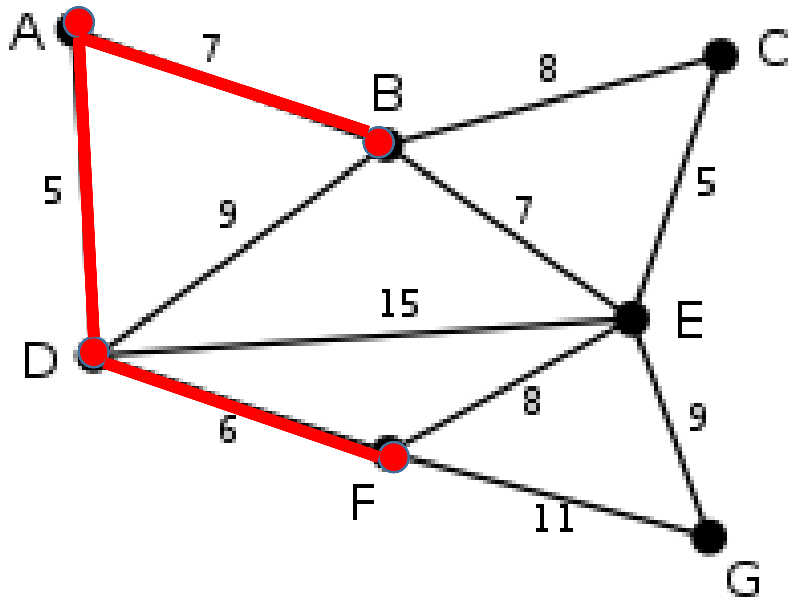
{B,C,E,G}

{D,A,F}

AB=7
DB=9
DE=15
FE=8
FG=11

$$E_{new} = \{DA, DF, AB\}$$

Example: Minimum spanning tree. Prim Algorithm



{C,E,G}

{D,A,F,B}

BC=8

BE=7

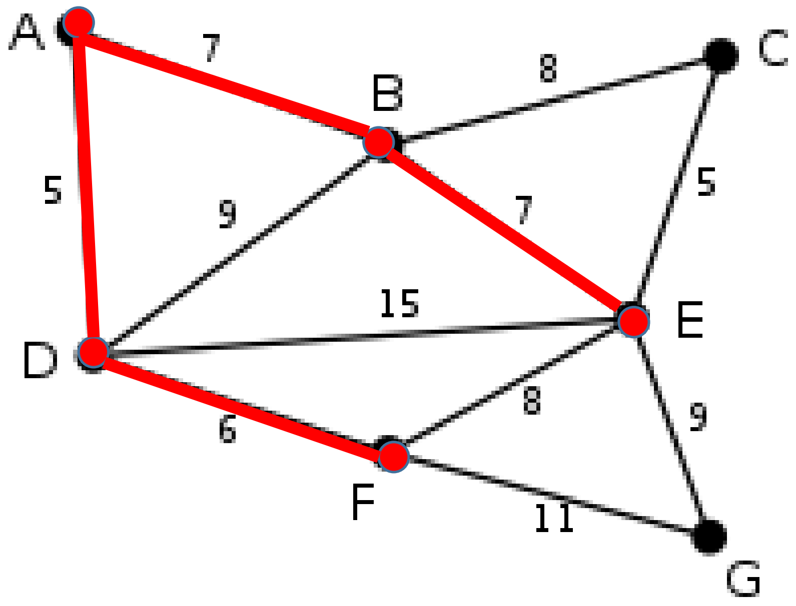
DE=15

FE=8

FG=11

$$E_{new} = \{DA, DF, AB, BE\}$$

Example: Minimum spanning tree. Prim Algorithm



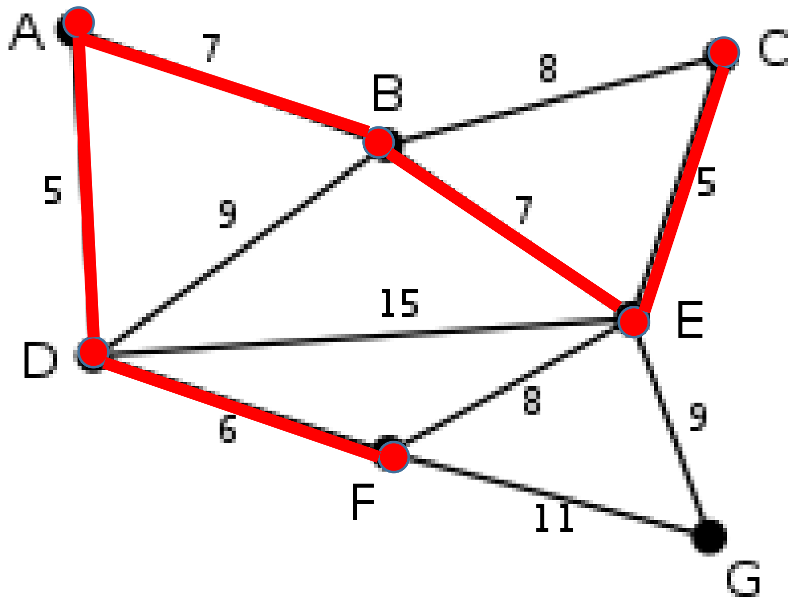
{C,G}

{D,A,F,B,E}

BC=8
EC=5
EG=9
FG=11

$$E_{new} = \{DA, DF, AB, BE, EC\}$$

Example: Minimum spanning tree. Prim Algorithm



{G}

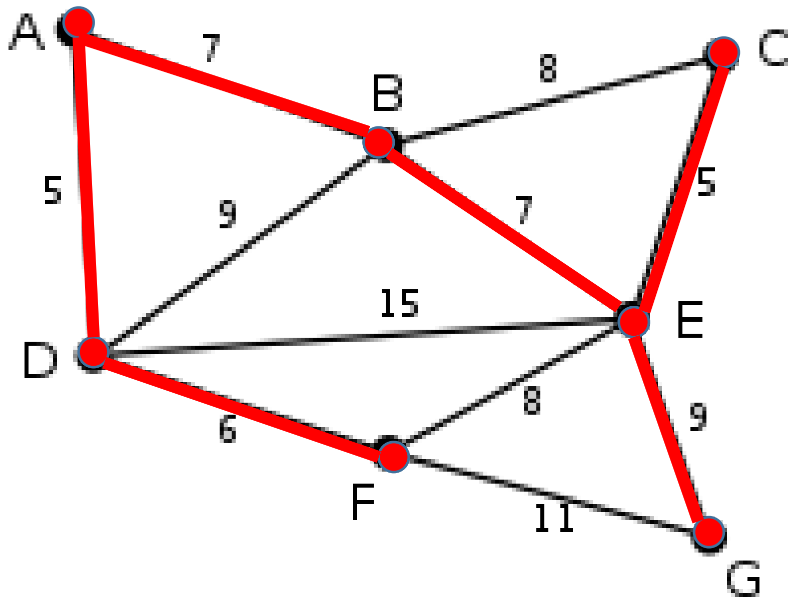
{D,A,F,B,E,C}

EG=9

FG=11

$$E_{new} = \{DA, DF, AB, BE, EC, EG\}$$

Example: Minimum spanning tree. Prim Algorithm

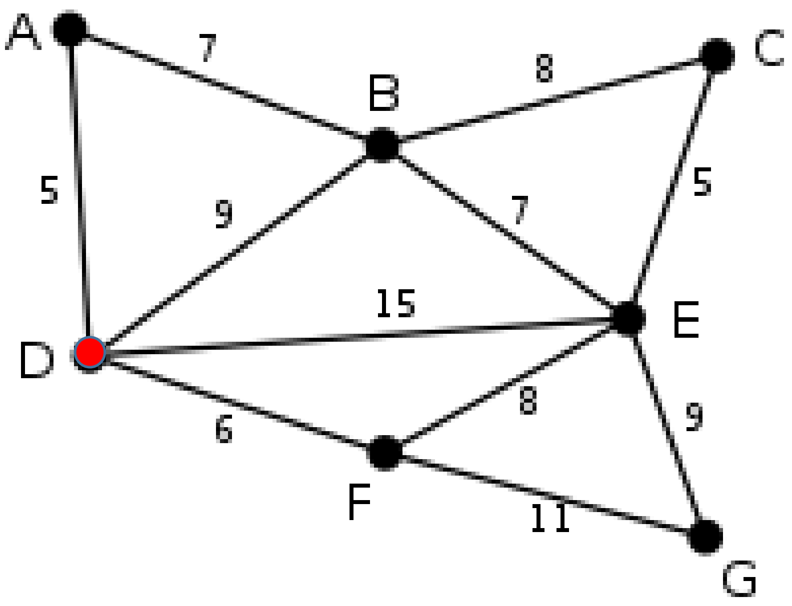


{}

{D,A,F,B,E,C,G}

$$E_{new} = \{DA, DF, AB, BE, EC, EG\}$$

Example: Minimum spanning tree. Prim Algorithm



$$
\begin{array}{c|ccccccc}
 & A & B & C & D & E & F & G \\
\hline
A & & & & & & & \\
B & & & & & & & \\
C & & & & & & & \\
D & & & & & & & \\
E & & & & & & & \\
F & & & & & & & \\
G & & & & & & &
\end{array}
$$

$$
\begin{array}{c}
A \\ B \\ C \\ D \\ E \\ F \\ G
\end{array}
\begin{pmatrix}
0 & 7 & +\infty & 5 & +\infty & +\infty & +\infty \\
7 & 0 & 8 & 9 & 7 & +\infty & +\infty \\
+\infty & 8 & 0 & +\infty & 5 & +\infty & +\infty \\
5 & 9 & +\infty & 0 & 15 & 6 & +\infty \\
+\infty & 7 & 5 & 15 & 0 & 8 & 9 \\
+\infty & +\infty & +\infty & 6 & 8 & 0 & 11 \\
+\infty & +\infty & +\infty & +\infty & 9 & 11 & 0
\end{pmatrix}
$$

Activity 5 Optimal control. Dynamic programming

Let's consider the system described by the dynamical model:

$$x_{k+1} = 2x_k + u_k$$

Considering an initial state $x_0 = 10$, obtain using a dynamic programming formulism the set of control signals u_k, $k = 0, \dots 9$, in a way that minimizes the cost function:

$$C = \sum_{k=0}^{9} \left(2 \cdot x_k^2 + u_k^2\right) + 2x_{10}^2$$

Represent graphically the vectors x_k and u_k as a function of k.

Dynamic programming **Example: Linear quadratic regulator**

Dynamic model

$$\boldsymbol{x}_{k+1} = A \cdot \boldsymbol{x}_k + B \cdot \boldsymbol{u}_k \qquad k \in \{0,1, \dots n-1\}$$

Cost function

$$C_{0\to h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2}\sum_{k=0}^{h.1}(\boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k) + \frac{1}{2}\boldsymbol{x}_h^T Q_h \boldsymbol{x}_h$$

$$J_{h-k}(\boldsymbol{x}_{h-k}) = \min_{u_{h-k}}\left(\frac{1}{2}\boldsymbol{x}_{h-k}^T Q \boldsymbol{x}_{h-k} + \frac{1}{2}\boldsymbol{u}_{h-k}^T R \boldsymbol{u}_{h-k} + J_{h-k+1}(\boldsymbol{x}_{h-k+1})\right)$$

$$\boldsymbol{u}_{h-k} = -\boxed{(R + B^T P_{h-k+1} B)^{-1} B^T P_{h-k+1} A} \cdot \boldsymbol{x}_{h-k} \qquad \boldsymbol{u}_{h-k} = -\boxed{K_{h-k}} \cdot \boldsymbol{x}_{h-k}$$

$$J_{h-k}(\boldsymbol{x}_{h-k}) = \frac{1}{2}\boldsymbol{x}_{h-k}^T \cdot \boxed{(Q + A^T P_{h-k+1} A - A^T P_{h-k+1} B \cdot (R + B^T P_{h-k+1} B)^{-1}\, B^T P_{h-k+1} A)} \cdot \boldsymbol{x}_{h-k}$$

$$J_{h-k}(\boldsymbol{x}_{h-k}) = \frac{1}{2}\boldsymbol{x}_{h-k}^T \cdot \boxed{P_{h-k}} \cdot \boldsymbol{x}_{h-k}$$

$$P_h = Q_h$$
$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B \cdot (R + B^T P_{k+1} B)^{-1}\, B^T P_{k+1} A$$
$$K_k = (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

$$\boldsymbol{u}_0 = -K_0 \cdot \boldsymbol{x}_0$$
$$\boldsymbol{x}_1 = A \cdot \boldsymbol{x}_0 + B \cdot \boldsymbol{u}_0$$

$$\boldsymbol{u}_1 = -K_1 \cdot \boldsymbol{x}_1$$
$$\boldsymbol{x}_2 = A \cdot \boldsymbol{x}_1 + B \cdot \boldsymbol{u}_1$$

...

Example: Linear quadratic regulator

Dynamic model

$$x_{k+1} = a \cdot x_k + b \cdot u_k \qquad k \in \{1,2, \dots 10\}$$

Cost function

$$C_{0\to h}(x_k, u_k) = \frac{1}{2}\sum_{k=1}^{10}(qx_k^2 + ru_k^2) + \frac{1}{2}q_{11}x_{11}^2$$

$$p_{11} = q_{11}$$

$$p_k = q + a^2 p_{k+1} - a^2 b^2 p_{k+1}^2 \cdot (r + b^2 p_{k+1})^{-1}$$

$$K_k = (r + b^2 p_{k+1})^{-1} a \cdot b \cdot p_{k+1}$$

$$x_1 = 10 \qquad u_1 = -K_1 \cdot x_1$$

$$x_2 = A \cdot x_1 + B \cdot u_1 \qquad u_2 = -K_2 \cdot x_2$$

$$x_3 = A \cdot x_2 + B \cdot u_2 \qquad u_3 = -K_3 \cdot x_3$$

$$\dots$$