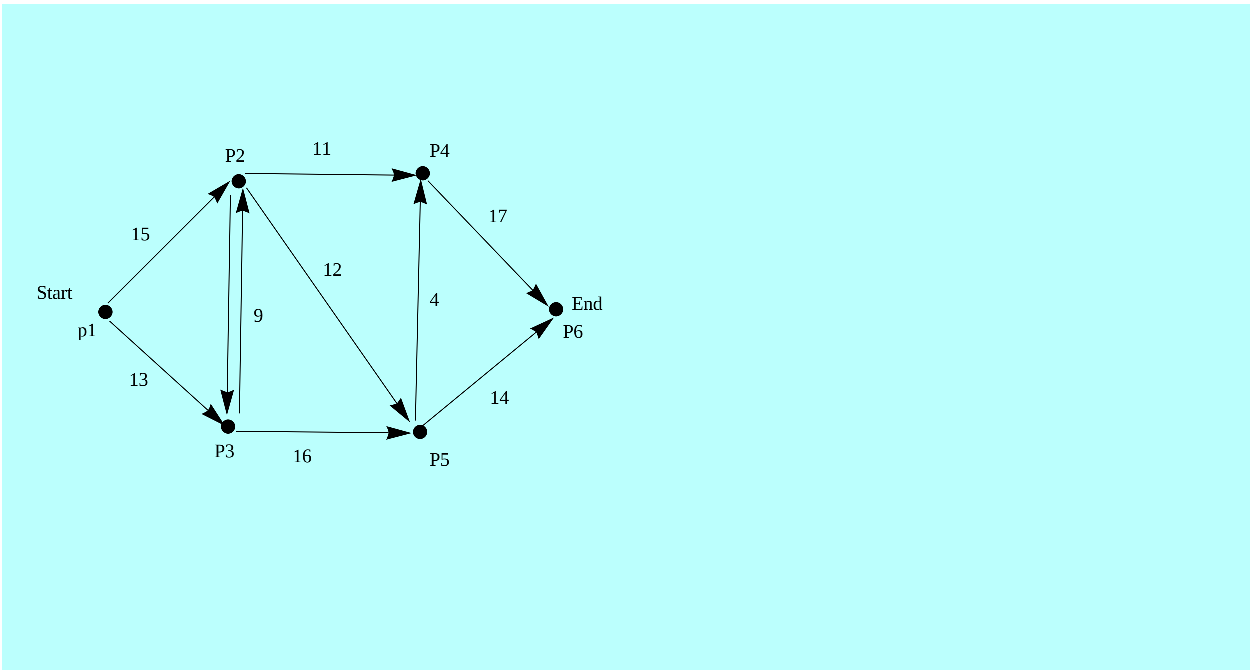
Shortest route problem

Find the shortest route from the starting point (p1) to the ending point (p6).



1. Set optimization variables

For each "link" in the graph, we set one variable. For example, there's a "link" from P1 to P2, so we set a variable x12. Notice that there are two variables associated with points P2 and P3, they are x23 and x32. If the value of a variable is 1, it means the "route" will pass through this link. Value 0 means this link will not be taken by the route.

In[1]:=
```
vars = {x12, x13, x23, x32, x24, x25, x35, x54, x46, x56};
```

2. Set the objective function

We would like to minimize the total "distance" (cost) of the chosen route :

In[2]:=
```
f = 15 * x12 + 13 * x13 + 9 * x23 + 9 * x32 +
   11 * x24 + 12 * x25 + 16 * x35 + 4 * x54 + 17 * x46 + 14 * x56;
```

3. Set the constraints

To make sure the route is a connected path from the starting point to the ending point, we need to following constrains:

- **(1) for each point other than the starting and the ending points, the total entering links (inflow)**

should be equal to the total leaving links (outflow).

In[3]:=
```
g2 = x12 + x32 == x24 + x25 + x23;
g3 = x13 + x23 == x32 + x35;
g4 = x24 + x54 == x46;
g5 = x35 + x25 == x54 + x56;
```

- **(2) for the starting point,   out flow - inflow = 1**

In[7]:=
```
g1 = x12 + x13 == 1;
```

- **(3) for the ending point, inflow - outflow = 1**

In[8]:=
```
g6 = x46 + x56 == 1;
```

- **(4) we also need to set all variables greater than or equal to 0**

In[9]:=
```
NonNegativeness = And @@ Thread[vars ≥ 0]
```

Out[9]=
```
x12 ≥ 0 && x13 ≥ 0 && x23 ≥ 0 && x32 ≥ 0 && x24 ≥ 0 && x25 ≥ 0 && x35 ≥ 0 && x54 ≥ 0 && x46 ≥ 0 && x56 ≥ 0
```

4. Solve the problem

- **(1) First, we can use the command "Minimize", which accepts equations/inequalities as inputs**

In[10]:=
```
Minimize[{f, g1 && g2 && g3 && g4 && g5 && g6 && NonNegativeness}, vars]
```

Out[10]=
```
{41, {x12 → 1, x13 → 0, x23 → 0, x32 → 0, x24 → 0, x25 → 1, x35 → 0, x54 → 0, x46 → 0, x56 → 1}}
```

The aboce solution tells us that the shortest route is   P1 -> P2 -> P5 -> P6, and the total cost is 41.

- **(2) *Mathematica* also provides a function "LinearProgramming", which is more efficient. However, it only accepts matrices/vectors in the standard LP form as inputs. Therefore, we first need to derive the coefficients in the standard form minimizing f = c^t x,   subject to constraints Ax >= b, x>=0. Notice that *Mathematica* uses Ax >= b, different from the Ax=b in the text book. Hence when we feed the command with inputs, we have to specify for each RHS value b_i that it is an exact "equal". This can be down by setting a matrix of the form {{b_1, 0}, {b_2, 0}, ... {b_k,0}}.**

In[11]:=
```
c = Normal[CoefficientArrays[f, vars] ] [[2]]
```

Out[11]=
```
{15, 13, 9, 9, 11, 12, 16, 4, 17, 14}
```

In[12]:=
```
A = Normal[CoefficientArrays[{g1, g2, g3, g4, g5, g6}, vars]] [[2]];
MatrixForm[A]
```

Out[13]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

In[14]:=
```
b = {{1, 0}, {0, 0}, {0, 0}, {0, 0}, {0, 0}, {1, 0}}
```

Out[14]=
```
{{1, 0}, {0, 0}, {0, 0}, {0, 0}, {0, 0}, {1, 0}}
```

In[15]:=
```
LPSol = LinearProgramming[c, A, b]
```

Out[15]=
```
{1, 0, 0, 0, 0, 1, 0, 0, 0, 1}
```

In[16]:=
```
LinearProgramming[c, A, b, Method → "Simplex"]
```

Out[16]=
```
{1, 0, 0, 0, 0, 1, 0, 0, 0, 1}
```

In[17]:=
```
LinearProgramming[c, A, b, Method → "RevisedSimplex"]
```

Out[17]=
```
{1, 0, 0, 0, 0, 1, 0, 0, 0, 1}
```

In[18]:=
```
LinearProgramming[c, A, b, Method → "InteriorPoint"]
```

```
LinearProgramming::lpipp :
 Warning: Method -> InteriorPoint specified for non-machine-precision
   problem. A machine-precision result will be given. If a
   non-machine-precision result is needed, set the option to Method -> Simplex.
```

Out[18]=

$\{1., 1.3007\times10^{-9}, 1.14458\times10^{-10}, 3.55423\times10^{-10}, 7.64473\times10^{-9}, 1., 1.05973\times10^{-9}, 2.27364\times10^{-10}, 7.8721\times10^{-9}, 1.\}$

Again, the above results indicate that the Shortest route is x12=1, x25=1, x56=1. The cost is

In[19]:=
```
f /. Thread[{x12, x13, x23, x32, x24, x25, x35, x54, x46, x56} → LPSol]
```

Out[19]=
```
41
```