Matrix determinant, inverse solving of system of equations

Matrix Determinant

In [ ]:
```
import numpy as np
from numpy import*
```

In [ ]:
```
# Creating a matrix
A = np.matrix('1 4 6; 2 56 54; 2 0 4')
A
```

Out[ ]:
```
matrix([[ 1,  4,  6],
        [ 2, 56, 54],
        [ 2,  0,  4]])
```

In [ ]:
```
# Calculating the determinant
det = np.linalg.det(A)
det
```

Out[ ]:
```
-47.9999999999999
```

In [ ]:
```
# Creatubg a matrix
C = np.matrix('1 2 5; 2 3 4; 5 6 8')
C
```

Out[ ]:
```
matrix([[1, 2, 5],
        [2, 3, 4],
        [5, 6, 8]])
```

In [ ]:
```
# Finding the inverse of the matrix
Inv = np.linalg.inv(C)
Inv
```

Out[ ]:
```
matrix([[ 1.77635684e-16, -2.00000000e+00,  1.00000000e+00],
        [-5.71428571e-01,  2.42857143e+00, -8.57142857e-01],
        [ 4.28571429e-01, -5.71428571e-01,  1.42857143e-01]])
```

Solving Linear Systems

In [ ]:
```
# Ax = B
a = np.matrix('3 -1 1;2 1 0; 1 2 -1')   #Co-efficients of x variable
b = np.matrix('2;1;3')                  #R.H.S value - B

#x = Inverse of A * B
x = np.linalg.inv(a) * b
print(x)
```

```
[[ 2.]
 [-3.]
 [-7.]]
```

x_1=2, x_2 = -3, x_3 = -7

In [ ]:
```
# Solving the linear system using linalg.solve()
ans = np.linalg.solve(a, b)
ans
```

Out[ ]:
```
matrix([[ 2.],
        [-3.],
        [-7.]])
```