HW1_DSA8420

January 25, 2025

1.) Compute following…

a.) $A + B$

[1]:
```
import numpy as np
from sympy import Matrix

A = np.array([[8, 7, -1], [-5, -3, 8], [4, 2, 6]])
B = np.array([[0, 3, -6], [2, 9, -5], [3, 4, 2]])

A_plus_B = A + B
A_plus_B
```

[1]:
```
array([[ 8, 10, -7],
       [-3,  6,  3],
       [ 7,  6,  8]])
```

b.) AB

[2]:
```
AB = np.dot(A, B)
AB
```

[2]:
```
array([[ 11,  83, -85],
       [ 18, -10,  61],
       [ 22,  54, -22]])
```

c.) $(BA)^T$

[3]:
```
BA = np.dot(B, A)
BA_T = BA.T
BA_T
```

[3]:
```
array([[-39, -49,  12],
       [-21, -23,  13],
       [-12,  40,  41]])
```

d.) tA

[4]:
```
t = -2
tA = t * A
```

```
tA
```

[4]:
```
array([[-16, -14,   2],
       [ 10,   6, -16],
       [ -8,  -4, -12]])
```

e.) Ax

[5]:
```
x = np.array([3, -1, 2])
Ax = np.dot(A, x)
print("Ax =", Ax)
```

```
Ax = [15  4 22]
```

f.) x^TB

[6]:
```
x_T = x.T
xT_B = np.dot(x_T, B)
print("xT_B =", xT_B)
```

```
xT_B = [ 4  8 -9]
```

2. Transform each of the following matrices to reduced row echelon form (RREF)

a.)

[7]:
```
A = Matrix([[1, -2, 3, 9], [-1, 3, 0, -4], [2, -5, 5, 17]])
A_rref, _ = A.rref()
A_rref
```

[7]:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

b.)

[8]:
```
B = Matrix([[1, 0, -2, 1, 0], [0, -1, -3, 1, 3], [-2, -1, 1, -1, 3], [0, 3, 9, 0, -12]])
B_rref, _ = B.rref()
B_rref
```

[8]:

$$\begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 3 & 0 & -4 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.) Find $(BA)^{-1}$

[9]:
```
A_inv = np.array([[3, 2], [1, 3]])
B_inv = np.array([[2, 3], [4, 1]])

BA_inv = np.dot(A_inv, B_inv)
BA_inv
```

```
[9]: array([[14, 11],
            [14,  6]])
```

4.) Find Invese of Matrices

a.)

```
[10]: matrix_a = np.array([[2, -2], [4, 3]])
      matrix_a_inv = np.linalg.inv(matrix_a)
      print("a^-1 =\n", matrix_a_inv)
```

```
a^-1 =
 [[ 0.21428571  0.14285714]
 [-0.28571429  0.14285714]]
```

b.)

```
[11]: matrix_b = np.array([[1, 2, 1], [1, 3, -1], [1, 2, 2]])
      matrix_b_inv = np.linalg.inv(matrix_b)
      print("b^-1 =\n", matrix_b_inv)
```

```
b^-1 =
 [[ 8. -2. -5.]
 [-3.  1.  2.]
 [-1.  0.  1.]]
```

c.)

```
[12]: matrix_c = np.array([[1, 0, 2], [2, 1, -1], [0, 4, 4]])
      matrix_c_inv = np.linalg.inv(matrix_c)
      print("c^-1 =\n", matrix_c_inv)
```

```
c^-1 =
 [[ 0.33333333  0.33333333 -0.08333333]
 [-0.33333333  0.16666667  0.20833333]
 [ 0.33333333 -0.16666667  0.04166667]]
```