Algorithm for QR Decomposition

Aim:

To implement QR decomposition algorithm using the Gram-Schmidt method.

Equipment's required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner

Algorithm:

1. Intialize the matrix Q and u
2. The vector u and e is given by

$$u_1 = a_1$$

$$u_2 = a_2 - (a_2.e_1)e_1$$

$$u_3 = a_3 - (a_3.e_1)e_1 - (a_3.e_2)e_2$$

$$e_1 = \frac{u_1}{\|u_1\|}$$

$$e_2 = \frac{u_2}{\|u_2\|}$$

3. Obtain the Q matrix

$$Q = (e_1|e_2| \dots \dots \dots e_n)$$

4. Construct the upper triangular matrix R

$$R = \begin{bmatrix} a_1.e_1 & a_2.e_1 & a_3.e_1 \\ 0 & a_2.e_2 & a_3.e_2 \\ 0 & 0 & a_3.e_3 \end{bmatrix}$$

Program:

Gram-Schmidt Method

```
'''
Program to QR decomposition using the Gram-Schmidt method
Developed by: Aldrin Lijo J E
RegisterNumber: 22008844
'''
import numpy as np
def QR_Decomposition(A):
    n,m =A.shape
    Q = np.empty((n,n))
    u = np.empty((n,n))
    u[:, 0] = A[:, 0]
    Q[:, 0]= u[:, 0] / np.linalg.norm(u[:, 0])
    for i in range(1, n):
        u[:, i] = A[:, i]
        for j in range(i):
            u[:, i] -=(A[:, i] @ Q[:, j])*Q[:, j]
        Q[:, i] = u[:, i]/np.linalg.norm(u[:, i])
    R = np.zeros((n, m))
    for i in range(n):
        for j in range(i, m):
            R[i, j] = A[:, j]@Q[:, i]
    print(Q)
```

```
    print(R)
a = np.array(eval(input()))
QR_Decomposition(a)
```

Output

Write the algorithm for QR decomposition using the Gram-Schmidt method.

For example:

Input	Result
([[1, 1, 0], [1,0,1], [0, 1, 1]])	[[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]

	Expected	Got	
0], [1,0,1], [0, 1, 1]])	[[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]	[[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]	✔
1, 4], [6, 167, -68], [-4, 24, -41]])	[[0.85714286 -0.39428571 -0.33142857] [0.42857143 0.90285714 0.03428571] [-0.28571429 0.17142857 -0.94285714]] [[14. 21. -14.] [0. 175. -70.] [0. 0. 35.]]	[[0.85714286 -0.39428571 -0.33142857] [0.42857143 0.90285714 0.03428571] [-0.28571429 0.17142857 -0.94285714]] [[14. 21. -14.] [0. 175. -70.] [0. 0. 35.]]	✔

✔

Result

Thus the QR decomposition algorithm using the Gram-Schmidt process is written and verified the result.