EE4013 ASSIGNMENT- 1

Shaik Abdur Rahman Nawaz - EE18BTECH11052

Download all codes from

```
https://github.com/AbdurNawaz/C−Data−
    Structures/tree/main/Assignment−1/codes
```

And Latex-tikz codes from -

```
https://github.com/AbdurNawaz/C−Data−
    Structures/tree/main/Assignment−1
```

1 Problem

Consider a matrix **P** whose only eigenvectors are the multiples of $\begin{bmatrix}1\\4\end{bmatrix}$.

Consider the following satements.

(i) **P** does not have an inverse.
(ii) **P** has a repeated eigenvalue.
(iii) **P** cannot be diagonalized.

Which of the following options is correct?

(A) only (i) and (iii) are necessarily true.
(B) only (ii) is necessarily true.
(C) only (i) and (ii) are necessarily true.
(D) only (ii) and (iii) are necessarily true.

2 Solution

We can observe that all the eigenvectors are linearly dependent.

Following are the properties/theorms that we will use,

1) Eigenvectors from different eigenvalues are linearly independent.
2) An n x n matrix is diagonalizable iff there are n linearly independent eigenvectors.

Sinice all the eigenvectors are linearly dependent, using property (1) we conclude that **P** has repeated eigenvalues.

Using property (2) we can conclude that **P** cannot be diagonalized.

Hence (D) is the right answer.

2.1 *Code Implementation*

We implement Gaussian elimination to convert the matrix(stack of all eigenvectors) to its row echelon form and find the rank of the matrix to check if all the eigenvectors are linearly independent.

Below is the code implemented to compute rank:

```
int compute_rank(float A[ROWS][COLS]) {

    int rank = COLS;

    for(int row=0;row<rank;row++){

        /*if diagonal element is non−zero,
        make all elements in the column
        0 except A[row][row].*/

        if(A[row][row]){

            for(int i=0;i<ROWS;i++){

                if(i!=row){
                    float temp = A[i][row]/A[
                        row][row];

                    for(int j=0;j<rank;j++)
                        A[i][j] −= temp*A[row
                            ][j];
                }
            }
        }

        /*else swap the row with any other
        row below it that has a non zero
        zero element in the same column.*/

        /*if there is non non zero element
        then just swap with the last row.*/

        else{
            bool flag = true;
            for(int i=row+1;i<ROWS;i++){
                if(A[i][row]){
                    flag = false;
                    swap(A, row, i, rank);
                    break;
                }
            }

            if(flag){
                rank−−;

                for(int i=0;i<ROWS;i++)
                    A[i][row] = A[i][rank];
            }

            row−−;
        }
    }

    return rank;
}
```

The complete code can be found at -

```
https://github.com/AbdurNawaz/C−Data−
    Structures/blob/main/Assignment−1/codes/
    ee18btech11052.c
```