1. Finding CUR of a small matrix

Find the *CUR* decomposition of the following matrix when we pick two "random" rows and columns as follows:

(1) If columns 0 and 1 and rows 1 and 2 are selected.

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

The following calculations must be in the derivation

(1) Computing the SVD of matrix W; say $W = X\Sigma Y^T$

(2) Computing Σ^+, the *Moore-Penrose pseudoinverse* of the diagonal matrix Σ

(3) Computing $U = Y(\Sigma^+)^2 X^T$

2. Comparing SVD, CUR and JL Transform

SVD, CUR decomposition and JL transform are two ways of reducing dimensionality to find the best approximation of the original data points using fewer dimensions. In this assignment, we compare these three dimension reduction techniques — SVD, CUR decomposition and JL Transform and we compare these three methods using reconstruction error (i.e. the squared distance between the original data and its "estimate").

Implement each of the above method which runs on a single machine (not distributed) in Java (preferably) or python. The input to the each program is the matrix as given in Question 1. Calculate the SVD, CUR and JL Transform of the input matrix for dimensions 1 to 5. Also, calculate the reconstruction error for both the methods for these dimensions. Tabulate the errors and plot them in a graph with increasing dimensions to show how the error varies.