```
- Sequential
    - Generate random matrix A and vector b of size passed through command line
    - Start Counting time
    - Loop overall all the rows
        - Do pivoting (find the maximum column element in this row and rows
          below and swap the rows if needed)
        - Divide the current row elements with the leading element, make leading
          element 1
        - Loop over all the rows beneath current row
            - Add negative leading element times above from all the rows in this
              loop (This will make sure everything beneath the leading element
              in current row becomes zero)
          End loop
      End loop
    - Loop from last row to start (Back substitution)
        - Populate the  (n-1)th entry in x vector using result from nth step
      End loop
    - Stop Counting time


- Parallel
- Generate random matrix A and vector b of size passed through command line
- Setup the threads
- Assign data to shared memory among threads
- Start counting time
- Loop overall all the rows
    - Do pivoting (find the maximum column element in this row and rows below and
      swap the rows if needed)
    - Divide the current row elements with the leading element, make leading element
    - Start n threads (Rows are not pre-divided)
        - Each thread executes while there are still rows left to be dealt
        - Put the Lock the current_row variable in current thread
            - Increment the current_row so next thread starts working on
              next row
        - Start working on the current row - 1
    - Wait for all threads to finish
  End loop

- Loop from last row to start (Back substitution)
    - Populate the  (n-1)th entry in x vector using result from nth step
  End loop
  Stop Counting Time
    -
-
```