```
Input: matrix l, matrix d, matrix u, array b,
        array x0, float tol, int n_max, int w
Output: matrix xact, array_steps E

begin sorMethod

        matrix T <- dot_product(inverse_matrix(d-dot_product(w,l)),
                (dot_product((1-w),d))+dot_product(w,u))
        matrix C <- dot_product((inverse_matrix(d-(w*l))*w),b.transpose())
        matrix C <- dot_product(inverse_matrix(d - l), b.transpose())
        float E <- infinity_value()
        array xant <- xo.transpose()
        int cont <- 0

        array values, array normalized_eigenvectors <- eigen_values(T)
        float spectral_radius <- maximum_value(absolute_value(values))

        while ((E > tol) and (cont < nmax))
                matrix xact <- dot_product(T, xant) + C
                E <- norm2(xant - xact)
                xant <- xact
                cont <- cont + 1
                array_steps[cont] <- xant
        end while

        xact, array_steps
endsorMethod
```