A Toeplitz matrix is an \( n \times n \) matrix \( A = (a_{ij}) \) such that \( a_{ij} = a_{i-1,j-1} \) for \( i = 2, 3, \ldots, n \) and \( j = 2, 3, \ldots, n \) (in other words, entries remain constant along each diagonal of the matrix).

Give an algorithm that multiplies an \( n \times n \) Toeplitz matrix by a vector of size \( n \) in \( O(n \log n) \) time. To achieve this, you need to think of an efficient way to represent the matrix, and you will have to use the FFT algorithm.

Write a program that implements the "generic" \( O(n^2) \) matrix-vector multiplication, and a program that implements your algorithm above. Run your code on random samples of size \( n = 10^k \) for \( k = 2, 3, 4, 5, 6 \). Write a report describing the results of your experiments.

You are strongly encouraged to code the entire assignment from scratch. However, you are allowed to use "off-the-shelf" implementations of the FFT algorithm, provided you disclose any source you use for the solution of this assignment.