Solved exersises for topics discussed in Lecture 6:

(FFT and convolution computation using FFT)

Schaum

273/7.1	Computational complexity using FFT
273/7.2	Computational complexity using FFT
273/7.3	Computational complexity using FFT
273/7.4	Computational complexity using FFT

Matlab (voluntary home-work, 1.5 points):

- Implement in Matlab the method for computation of convolution named **Overlap-Save** and compare its computational complexity with Matlab command *conv()*. For the comparison, use long signal (N=1e6 and more samples) and long impulse response (L=2000 and more). The signals and implulse responses can be generated randomly, using Matlab command *randn()*. Select a suitable length of the overlap-save block, such that the computation is faster than the *conv()* command and simultaneously the lag is smaller than 10000 samples.
- Verify:
  - the time-effectivness of your code using the time-measuring commands *tic/toc*. Matlab automatically paralelizes some of its computations, it is neccesary, for a fair comparisson, to restrict the allowed number of threads to 1 (command *maxNumCompThreads*).
  - using the squared error *sum((x1-x2).^2)* that both implementations of convolution give the same result (the error will be non-zero due to round-off errors but it should be negligible).

- Evaluation criteria
  - Functionality/runable + correctness of solution
  - Comments in code
  - Originality
  - Effective code (fast)