1. [7] Let \( H[1:n] \) be an array of integer values in the range \( 1 \ldots k \), with that \( 1 \leq k \leq n \). We want to find the most frequently occurring value \( m \) in \( H \), i.e. the mode of \( H \). For example, with \( n = 8, k = 4 \), and \( H = [1, 3, 1, 3, 3, 4, 3, 1] \), we should find \( m = 3 \). For simplicity you can assume that the mode is unique.

   **(a)** Verify that the sequential time complexity for this problem is \( \Theta(n) \).

   **(b)** Describe an efficient parallel BSP algorithm for this problem using \( p \) processors assuming the condition \( n = kp \) with \( k \geq p \) and give its BSP cost. The input \( H \) is distributed evenly over processors, so that initially each processor holds \( k \) input values (assume \( p \) divides \( n \) evenly). The result \( m \) should be available in the first processor on termination.

2. [10] The Discrete Fourier Transform (DFT) of a sequence of complex values \( X[0:n-1] \) where \( n = 2^r \) yields complex values \( Y[0:n-1] \) (where \( Y_i = \sum_{0 \leq k < n} X_k \omega^{ki} \) and \( \omega = e^{2\pi\sqrt{-1}/n} \)). The radix-2 Fast Fourier Transform (FFT) computes \( Y[0:n-1] \) (in bit-reversed index order) and can be expressed as a W-T model parallel algorithm with \( S(n) = O(lg n) \) and \( W(n) = O(n \ lg n) \) as follows

   ```
   forall i \in 0:n-1 do
     Y[i] := X[i]
   end
   for m := 0 to r-1 do
     forall i \in 0:n-1 do
       let \((b_0 \ldots b_{m-1} b_m b_{m+1} \ldots b_{r-1})\) be the binary representation of \( i \)
       int j := \((b_0 \ldots b_{m-1} 0 b_{m+1} \ldots b_{r-1})\)
       int k := \((b_0 \ldots b_{m-1} 1 b_{m+1} \ldots b_{r-1})\)
       int h := \((b_m \ldots b_{m-1} 0 0 \ldots 0)\)
       Y[i] := Y[j] + Y[k] \cdot \omega^h
     end forall
   end for
   ```

   **(a)** Construct an algorithm in the BSP model where \( n = p \) with BSP cost \( O(lg p)(1 + g + L) \).

   **(b)** Construct an algorithm in the BSP model where \( n = 2^r \) and \( r \geq 2 \ lg p \) with BSP cost

   \[
   O(1) \left( \frac{n \ lg n}{p} + \frac{n}{p} \cdot g + L \right)
   \]

   In both cases \( X \) and \( Y \) should be distributed evenly over processors.