Generating elementary combinatorial objects

1. (30 points) Correctness of Successor algorithm for Graycodes
   Prove Theorem 2.2 of the textbook which states that Algorithm 2.3 (also given in class) correctly computes successor for the binary reflected Gray code. You need to state and prove several facts, which will formalize the informal statements given in page 38 of the textbook:

   Algorithm 2.3 works as follows. If \( w(A) \) is even, then the last bit of \( A \) (namely \( a_0 \)) is flipped; if \( w(A) \) is odd, then we find the first “1” from the right, and flip the next bit (to the left).

   The last vector in \( G^n \), which has no successor (added by editor: or successor \([0, \ldots, 0]\) in circular order) is \([1, 0, \ldots, 0]\) This corresponds to the set \{1\}.

   Hint: Prove the facts by induction on \( n \).

2. (30 points) Generating all variations of a multiset
   The variations of a multiset are the permutations of all its sub-multisets.

   Example: the variations of multiset \([1, 2, 2, 3]\) are:

   \[
   \epsilon, 1, 12, 122, 1223, 123, 13, 132, 1322 \quad 2, 21, 212, 2123, 213, 22, 221, 2213, 223, 23, 231, 2312, 232, 2321 \quad 3, 31, 312, 3122, 32, 321, 3212, 322, 3221
   \]

   Give an algorithm (pseudocode) to generate all variations of a multiset.

   To verify that your algorithm works, you may implement it and test it; however the implementation is not required for the assignment.

   Hint: You may use a similar idea as the one for lexicographical order of permutations.

3. (40 points) Generalized Gray codes
   (a) Let \( m_0, m_1, \ldots, m_{n-1} \) be integer numbers greater than or equal 2. In this exercise we want to generate all \( n \)-tuples \((a_{n-1}, a_{n-2}, \ldots, a_1, a_0)\) where \( 0 \leq a_j < m_j \) for all \( j, 0 \leq j < n \), according to the following minimal change ordering: two successive tuples differ in exactly one component with the absolute value of their difference equals to 1 (i.e. the component is either incremented or decremented by 1). Adapt the binary reflected Gray code successor algorithm to the case of this generalized Gray code. Give your algorithm in pseudocode form.

   (b) Given the prime factorization of a number \( p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t} \), give an algorithm to run through all divisors of the number, by repeatedly multiplying or dividing by a single prime at each step.

   Hint: Use the algorithm developed in part 1.