FFT and minimum spanning trees

1. Given a longer string s (of length n) and a shorter string t (of length m) over a binary alphabet. For each i, calculate the similarity of $s[i...(i + m - 1)]$ and t. The similarity of two strings is defined as the number of positions where they have the same character. For example, the similarity of "abbaa" and "aabba" is 3, because they match on positions 0, 2 and 4.
2. Suppose you have n rocks and the i-th of them weighs a_i grams. How many ways are there to pick rocks such that their total weight is exactly k? Depending on the constraints and the specific problem, there are several ways to approach this. But one way to model them is with polynomials. The answer to the question is exactly the coefficient of x^k in the polynomial $(1 + x^{a_0})(1 + x^{a_1}) \cdots (1 + x^{a_{n-1}})$. What is the time required to compute this?
3. This problem illustrates how to do the Fourier Transform (FT) in modular arithmetic, for example, modulo 7.
    a. There is a number ω such that all the powers ω, ω^2, ω^3 ..., ω^6 are distinct (modulo 7). Find this ω, and show that $\omega + \omega^2 + \omega^3 + \cdots + \omega^6 = 0$. (Interestingly, for any prime modulus there is such a number.)
    b. Using the matrix form of the FT, produce the transform of the sequence (0, 1, 1, 1, 5, 2) modulo 7; that is, multiply this vector by the matrix $M_6(\omega)$ for the value of ω you found earlier. Recall the (j, k) entry of this matrix is ω^{jk} where $1 \leq j, k \leq 6$. In the matrix multiplication, all calculations should be performed modulo 7.
    c. Write down the matrix necessary to perform the inverse FT. Show that multiplying by this matrix returns the original sequence. (Again, all arithmetic should be performed modulo 7.)
    d. Now show how to multiply the polynomials $x^2 + x + 1$ and $x^3 + 2x - 11$ using the FT modulo 7.
4. Suppose you are given an undirected graph G, with edge weights that you may assume are all distinct. G has n vertices and m edges. A particular edge e of G is specified. Give an algorithm with running time $O(m + n)$ to decide whether e is contained in a minimum-weight spanning tree of G.
5. Consider Boruvka's algorithm for finding a minimum spanning tree and prove its correctness.
    a. Initialize each vertex in the graph as its own separate component (or set).
    b. Initialize the MST as an empty set of edges
    c. While there is more than one component remaining in the graph:
        i. For each current component, identify the cheapest edge that connects this component to a different component.
        ii. Add all such identified cheapest edges to the MST. If an edge connects two components that are already merged in the current iteration, it is not added again.
        iii. Merge the components that are connected by the newly added edges.

d. The algorithm terminates when only one component remains. The set of edges accumulated in the MST forms the Minimum Spanning Tree.

.