Divide-and-conquer algorithms

In this approach, a large computational problem is divided into smaller computational problems, where each smaller problem is tackled separately by the algorithm (for example, by using recursion).

Example: merge sort (see, for example, https://www.geeksforgeeks.org/merge-sort/).

In merge sort, the goal is to sort the elements (for example, numbers) in the array (for example, in non-decreasing order). The array is divided into two halves, each half is sorted separately by using recursion, and the resulting sorted halves are “merged” together.

Fast multiplication of polynomials

Let

\[ A(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{m} a_i x^i \]

and

\[ B(x) = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_1 x + b_0 = \sum_{i=0}^{m} b_i x^i \]

be two polynomials with real coefficients of degree \( m \) (some leading coefficients can be zero). Our goal is to efficiently compute the polynomial \( C(x) = A(x) \cdot B(x) \). Denote

\[ C(x) = b_{2m} x^{2m} + b_{2m-1} x^{2m-1} + \cdots + b_1 x + b_0 = \sum_{i=0}^{2m} b_i x^i , \]

where \( c_k = \sum_{i=0}^{k} a_i b_{k-i} \).

Computing this polynomial in a straightforward manner will require time \( O(m^2) \). In this and the next lecture, we will study a method that works with lower complexity.

**Theorem 1** A polynomial of degree \( m \) is uniquely characterized by its values at any \( m + 1 \) distinct points.

Let the points \( x_0, x_1, \ldots, x_m \) be fixed. There is a bijection between the coefficients \( a_0, a_1, \ldots, a_m \) and the values of \( A(x_0), A(x_1), \ldots, A(x_m) \).

Observe that if for any \( i = 0, 1, \ldots, m \), the values of \( A(x_i) \) and \( B(x_i) \) are known, then it is very easy to compute all \( C(x_i) = A(x_i) \cdot B(x_i) \). This gives us the idea of the following algorithm.
Algorithm: fast multiplication of polynomials

**Input:** Polynomials $A(x) = \sum_{i=0}^{m} a_i x^i$ and $B(x) = \sum_{i=0}^{m} b_i x^i$.

**Output:** Polynomial $C(x) = \sum_{i=0}^{2m} c_i x^i$, such that $C(x) = A(x) \cdot B(x)$.

1. **Point selection.** Select distinct points $x_0, x_1, \cdots, x_{n-1}$, $n \geq 2m + 1$.
2. **Evaluation.** Compute $A(x_0), A(x_1), \cdots, A(x_{n-1})$ and $B(x_0), B(x_1), \cdots, B(x_{n-1})$.
3. **Multiplication.** For $i = 0, 1, \cdots, n-1$, compute $C(x_i) = A(x_i) \cdot B(x_i)$.
4. **Interpolation.** Recover $C(x) = \sum_{i=0}^{n-1} c_i x^i$. 
