Algorithms and Complexity. Exercise session 2

Greedy + Divide and Conquer

MST greedy Describe a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. Namely, it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.

Max and min Let $v[1..n]$ be a vector of n integers. Give a recursive algorithm that computes the largest and the smallest number in v. The algorithm uses at most $\lceil 3n/2 \rceil - 2$ comparisons between elements in v. Note that the number of elements in v may not be a power of 2.

Matrix multiplication Strassen's algorithm multiplies two $n \times n$-matrices in time $O(n^{2.808})$ by decomposition of 2×2-block matrices. It is faster than $O(n^3)$ because it computes seven multiplications instead of eight to form the product matrix. Another idea is to make the decomposition of 3×3-block matrices instead. A researcher at NADA tried a couple of years ago to find the minimum number of multiplications needed to multiply two 3×3-matrices. He managed to get almost 22 multiplications. If he had succeeded, what would have been the time complexity to the multiplication of two $n \times n$-matrices?

Complex multiplication If we multiply two complex numbers $a + bi$ and $c + di$ in the standard way, it requires four multiplications and two additions of real numbers. Since multiplications are more expensive than additions (and subtractions), it pays to minimize the number of multiplications if one would allow large numbers. Find an algorithm that uses only three multiplications (but more additions) to multiply two complex numbers.

Majority Consider an array A of n elements (say integers). Construct and analyze an algorithm to determine whether any element of the array A is in majority, namely, it occurs in A at least $n/2$ times. If this is the case, return it. The algorithm will be a recursive one and will have time complexity $O(n \log n)$. The only operation you are allowed to use on the elements of A is $=$ (equality test). Moreover, there is no order relationship between the elements.

Inside or outside? Let P be a convex n-angle polygon described as an array of angles $p_1, p_2, \ldots p_n$ in cyclic order. Construct an algorithm that computes whether a given point q is inside the polygon P. The algorithm will run in time $O(\log n)$ in worst case.
