Final Exam
(due in class on Monday)

This is the usual open-everything, but no outside help take-home test. Check "Class News", where I will post “frequently asked questions” about the test.

1. Amortized Complexity of Union – Find

Here we assume that we have a partition of \( n \) elements into disjoint sets, where each set is represented by a tree in which non-root nodes have pointers to their parent nodes.

(i) Show that Find with path compression alone (without a smarter Union than an arbitrary link of roots) has \( \Omega(\log n) \) amortized complexity.

(ii) Show that in the implementation as in (i) (with an arbitrary link), if all Union operations precede any Find operation, then the amortized complexity of Find is constant.

2. Greedy Loop Invariant

This problem pertains to the construction of an optimal prefix-free binary code for \( n \) messages by the Huffman algorithm. Assume that the probabilities of message transmission are given in the non-decreasing order: \( p_1 \leq p_2 \leq \cdots \leq p_n \). (An essential assumption!)

(i) Prove the following invariant of the algorithm constructing a binary tree representing an optimal code: The pseudo-messages that represent the parents of deleted lowest frequencies nodes are created and removed in nondecreasing order of their frequencies.

("Pseudo-messages” are the internal nodes of the tree, of which the messages are leaves.)

(ii) Write a pseudocode implementing the construction algorithm to perform in linear time while maintaining the above invariant. Prove its correctness. Make sure to describe a data structure taking advantage of the invariant.
3. Divide and Conquer

Analyze correctness and complexity of the following algorithm for order statistics algorithm selecting the $k^{th}$ smallest out of $n$ elements $a_1, a_2, \ldots, a_n$:

(i) Select the middle (second smallest) element $m_i$ of each triple of numbers $(a_1, a_2, a_3), \ldots, (a_{n-2}, a_{n-1}, a_n)$;

(ii) Recursively find the median ($\frac{n}{6}$th smallest) element $M$ of these $\frac{n}{3}$ numbers;

(iii) Partition the original $n$ numbers into those smaller than $M$, say $b_1, b_2, \ldots, b_m$ and the rest $b_{m+1}, \ldots, b_n$;

(iv) If $k \leq m$ then recursively find the $k^{th}$ smallest among $b_1, b_2, \ldots, b_m$, otherwise find the $(k - m)^{th}$ smallest among $b_{m+1}, \ldots, b_n$.

4. Fancy Fourier

Not really FFT but something closely related.

(i) Describe a result of multiplying two expressions: $A(x) = q_1x^1 + q_2x^2 + \ldots + q_nx^n$ and $B(x) = \frac{1}{n^2}x^n + \frac{1}{(n-1)^2}x^{n-1} + \ldots + \frac{1}{4}x^2 + x - x^{-1} - \frac{1}{4}x^{-2} - \ldots - \frac{1}{(n-1)^2}x^{-n+1} - \frac{1}{n^2}x^{-n}$ (a general coefficient of the resulting expression will suffice.)

(ii) Using the result of (i) above, solve the problem 5.4 (page 247 in text.) Again, a high level description with some details will be better than a piece of code.)