This is the usual open-everything, no web search and no outside help take-home test. Provide answers with their justifications. Please staple this page in front of your solutions.

Check Piazza, where I will post “frequently asked questions” about the test – but communicate those questions to me only, via email.

1. Invariants

1. Let the array Coeff[0..n] hold the coefficients $a_i$ of a polynomial $p(x) = \sum_{0 \leq i \leq n} a_i x^i$. Prove by loop invariant that the following program evaluates $p(x)$ at $x = \text{Var}$.

   Poly := 0; i := n;
   while $i \geq 0$ do {Poly := Poly*Var + Coeff[i]; i := i-1}

2. Prove by loop invariant the correctness of the following algorithm sorting an array A[1..n]:

   i := n;
   while $i > 1$ do
   {j := 1;
    i := i-1}
2. Algorithms

1. Show all the computational steps needed to evaluate the convolution of 
   \((2 -1 4 3)\) and \((1 -3 2 4)\) using the FFT technique. (Hint: \(\omega(1,8) = (1 + i)\sqrt{2}/2\).)

2. Input is a binary tree \(T\) with weighted nodes. Describe a linear time algorithm for selecting a subset \(W\) of the nodes with maximum total weight but subject to the condition that \(W\) does not contain any pair of nodes \(v, w\) in which \(w\) is a child of \(v\).

3. Computing the product of two rectangular matrices \(p \times q\) and \(q \times r\) takes \(pqr\) scalar multiplications. The matrix chain product problem asks for the minimum number of scalar multiplications required to compute the product of \(n\) matrices with dimensions \(p_0 \times p_1, p_1 \times p_2, \ldots, p_{n-1} \times p_n\).
   (i) Solve the matrix chain product problem for the chain of 5 matrices with dimensions \(100 \times 50, 50 \times 2, 2 \times 100, 100 \times 50, 50 \times 3\), respectively. Your solution should show the recursive structure of the problem and include the computations leading to filling of the matrix of optimal solutions to subproblems (form at the bottom of this page).
   (ii) Indicate the parenthesization of \(M_1 \times M_2 \times M_3 \times M_4 \times M_5\) that results in the minimum number of scalar multiplications found in (i) above.

3. Abstract complexity

1. Show that the polynomial-time reduction, \(\leq_p\), defines a transitive relation between problems.

2. N. (name withheld) claims that he can solve any SAT problem in polynomial time. He claims to have an algorithm that can do this. Can you use N’s algorithm to efficiently find a satisfying assignment?

3. Give an instance of the Hamiltonian Cycle problem corresponding (in the reduction from VC discussed in class) to the instance \((C_4,3)\) of the Vertex Cover problem. \((C_4\) is the cycle of four vertices, 3 is the size of vertex cover.) For both instances, give the evidence supporting the answers. (Hint: The construction is illustrated in the figure below taken from Gary and Johnson.)

![Figure 3.4](image-url)