"Is there anything in the Geneva Convention about the rules of war in peacetime?" Stanko wanted to know, crawling back toward the truck.


– Gasp, Romain Gary

**Required Problems**

1. **Sorting networks stuff** (40 pts.)
   
   (A) (5 pts.) Prove that an $n$-input sorting network must contain at least one comparator between the $i$th and $(i+1)$st lines for all $i = 1, 2, ..., n-1$.
   
   (B) (20 pts.) Prove that in a sorting network for $n$ inputs, there must be at least $\Omega(n \log n)$ gates. For full credit, your answer should be short, and self contained (i.e., no reduction please).

   [As an exercise, you should think why your proof does not imply that a regular sorting algorithm takes $\Omega(n \log n)$ time in the worst case.]

   (C) (5 pts.) Suppose that we have $2n$ elements $\langle a_1, a_2, ..., a_{2n} \rangle$ and wish to partition them into the $n$ smallest and the $n$ largest. Prove that we can do this in constant additional depth after separately sorting $\langle a_1, a_2, ..., a_n \rangle$ and $\langle a_{n+1}, a_{n+2}, ..., a_{2n} \rangle$.

   (D) (10 pts.) Let $S(k)$ be the depth of a sorting network with $k$ inputs, and let $M(k)$ be the depth of a merging network with $2k$ inputs. Suppose that we have a sequence of $n$ numbers to be sorted and we know that every number is within $k$ positions of its correct position in the sorted order, which means that we need to move each number at most $(k-1)$ positions to sort the inputs. For example, in the sequence 3 2 1 4 5 8 7 6 9, every number is within 3 positions of its correct position. But in sequence 3 2 1 4 5 9 8 7 6, the number 9 and 6 are outside 3 positions of its correct position.

   Show that we can sort the $n$ numbers in depth $S(k) + 2M(k)$. (You need to prove your answer is correct.)

2. **Computing Polynomials Quickly** (30 pts.)

   In the following, assume that given two polynomials $p(x), q(x)$ of degree at most $n$, one can compute the polynomial remainder of $p(x) \mod q(x)$ in $O(n \log n)$ time. The **remainder** of $r(x) = p(x) \mod q(x)$ is the unique polynomial of degree smaller than this of $q(x)$, such that $p(x) = q(x) \cdot d(x) + r(x)$, where $d(x)$ is a polynomial.

   Let $p(x) = \sum_{i=0}^{n} a_i x^i$ be a given polynomial.

   (A) (8 pts.) Prove that $p(x) \mod (x-z) = p(z)$, for all $z$. 

1
(B) (8 pts.) We want to evaluate \( p(\cdot) \) on the points \( x_0, x_1, \ldots, x_{n-1} \). Let

\[
P_{ij}(x) = \prod_{k=i}^{j} (x - x_k)
\]

and

\[
Q_{ij}(x) = p(x) \mod P_{ij}(x).
\]

Observe that the degree of \( Q_{ij} \) is at most \( j - i \).
Prove that, for all \( x \), \( Q_{kk}(x) = p(x_k) \) and \( Q_{0,n-1}(x) = p(x) \).

(C) (6 pts.) Prove that for \( i \leq k \leq j \), we have

\[
\forall x \quad Q_{ik}(x) = Q_{ij}(x) \mod P_{ik}(x)
\]

and

\[
\forall x \quad Q_{kj}(x) = Q_{ij}(x) \mod P_{kj}(x).
\]

(D) (8 pts.) Given an \( O(n \log^2 n) \) time algorithm to evaluate \( p(x_0), \ldots, p(x_{n-1}) \). Here \( x_0, \ldots, x_{n-1} \) are \( n \) given real numbers.

3. Linear time Union-Find. (30 pts.)

(A) (3 pts.) With path compression and union by rank, during the lifetime of a Union-Find data-structure, how many elements would have rank equal to \( \lfloor \lg n - 5 \rfloor \), where there are \( n \) elements stored in the data-structure?

(B) (3 pts.) Same question, for rank \( \lfloor (\lg n)/2 \rfloor \).

(C) (6 pts.) Prove that in a set of \( n \) elements, a sequence of \( n \) consecutive FIND operations take \( O(n) \) time in total.

(D) (3 pts.) Write a non-recursive version of FIND with path compression.

(E) (9 pts.) Show that any sequence of \( m \) MAKESET, FIND, and UNION operations, where all the UNION operations appear before any of the FIND operations, takes only \( O(m) \) time if both path compression and union by rank are used.

(F) (6 pts.) What happens in the same situation if only the path compression is used?