B1. Algorithms (25 points)

Search Trees

a) (7 points) Define a tree data structure that requires at most $\Theta(\lg n)$ time to insert or find an element in a tree containing n items. Draw a picture of such a tree.

b) (9 points) Let t_o denote the empty tree of type (a) and let t_i be the tree t_{i-1} with the integer i inserted. Draw the eight trees $t_1, t_2, \ldots, t_8$.

c) (9 points) Give the analysis that shows that an insert or find operation requires at most $\Theta(\lg n)$ time.

B2. Algorithms (25 points)

Minimum Spanning Trees

Let (G, w) be a weighted graph, (i.e., G = (V, E), $w : E \longrightarrow \mathbb{R}$).

a) (4 points) Define minimum spanning tree.

Consider this algorithm:

```
Generic-MST(G, w)
1.  A <-- Ø
2.  While A is not a spanning tree do
    2.1  find an edge (u, v) that is "safe" for A
    2.2  A <-- A ∪ {(u, v)}
3.  Return A.
```

b) (6 points) Explain Kruskal's algorithm as an elaboration of the above. Describe the meaning of "safe" used in Kruskal's algorithm and describe the data structure you would use to make finding safe edges efficient.

c) (3 points) Give a good bound for the run time in terms of E and V . By abuse of notation, you may use E and V to denote their sizes E and V .

d) (6 points) Repeat part (b) for Prim's algorithm.

e) (3 points) Repeat part (c) for Prim's algorithm.

f) (3 points) Name and briefly describe an application that would use a minimum spanning tree algorithm.

B3. Algorithms (25 points)

Polynomial Multiplication Using Divide and Conquer

Let $N = 2n$ be a power of two (for simplicity).

Let $A = a_{2n-1}\, x^{2n-1} + \ldots + a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$

$B = b_{2n-1}\, x^{2n-1} + \ldots + b_n x^n + b_{n-1}\, x^{n-1} + \ldots + b_1 x + b_0$

be two polynomials with deg $A < N$ and deg $B < N$.

Then we can write $A = A_1(x).x^n + A_0(x)$

$B = B_1(x).x^n + B_0(x)$

where A_1, A_0, B_1, B_0 are polynomials each with degrees $< n = N/2$. Now note that

$$A.B = (A_1 x^n + A_0)\ (B_1 x^n + B_0) \qquad (1)$$

$$= A_1 B_1 x^{2n} + (A_1 B_0 + A_0 B_1)\ x^n + A_0 B_0$$

and $(A_1 B_0 + A_0 B_1) = (A_1 + A_0)(B_1 + B_0) - A_1 B_1 - A_0 B_0$ (2)

a) (15 points) Use (1) and (2) to obtain an algorithm that runs in less than $\mathcal{O}(N^2)$ time to multiply two polynomials A + B each of degree $< N$ by carrying out only three multiplications of polynomials of degree $< n = N/2$ plus operations that are linear in N. Present the algorithm in pseudo-code. You may assume a linear algorithm PADD for addition of two polynomials.

b) (10 points) What is the worst case computing time of your algorithm. Justify your answer.

B4. Algorithms (25 points)

Finding a maximum on a PRAM.

a) (5 points) Explain how to find the maximum of n numbers in $\Theta(\log(n))$ time on a PRAM. You may use the Prefix Sums Algorithm.

b) (5 points) What is a common CRCW PRAM?

c) (10 points) Give an algorithm in pseudo-code to compute the maximum of n numbers in $\Theta(1)$ time on a common CRCW PRAM.

d) (5 points) Explain how accelerated cascading may be used with (a) and (c) to obtain an optimal algorithm for maximum that runs in $\Theta(\log \log n)$ time.
[Hint: It will help to develop a non-optimal algorithm from (c) that runs in $\Theta(\log \log n)$ time.]