FUNDAMENTAL ALGORITHMS
OLD MIDTERM

No calculators, no notes. Do all problems. Maximal score: 125.

1. (20) Consider an algorithm HULYE for multiplication of two \( n \) digit numbers. (Don’t worry how HULYE really works, we just want an analysis given the information below.) It multiplies two \( n \) digit numbers by making six recursive calls to multiplication of two \( n/2 \) digit numbers plus two additions of \( n \) digit numbers. Each of the additions take time \( O(n) \). Give the recursion for the time \( T(n) \) for HULYE and use the Master Theorem to find the asymptotics of \( T(n) \). Is HULYE a good algorithm to use for \( n \) large? Give a brief reason for your answer.

2. (25) Consider the following Binary Search Tree TREE with \( \text{ROOT(TREE)} = K \). (The values have been deliberately excluded. Assume the values are distinct.)

```
 vertex  Q  I  R  S  U  T  C  K  O
 leftchild  NIL  U  O  NIL  NIL  NIL  Q  NIL
 rightchild  I  C  T  R  NIL  NIL  S  NIL
 parent  K  Q  S  K  I  R  I  NIL  R
```

(a) (5) Draw a (nice!) picture of this tree.
(b) (10) Which is the vertex with minimal value. Illustrate how the program MIN will find it.
(c) (10) Give the vertices of the tree in increasing order of value. (Give an indication of your method, but you needn’t give every detail.)

3. (20) Let \( A \) be an array of length 127 in which the values are distinct and in increasing order.

(a) In the procedure BUILD-MAX-HEAP(\( A \)) precisely how many times will two elements of the array be exchanged? (Reason, please!)
(b) Now suppose the values are distinct and in decreasing order. Again, in the procedure BUILD-MAX-HEAP(\( A \)) precisely how many times will two elements of the array be exchanged? (Reason, please!)
4. (20) Let $V[1 \cdots N]$ be an array of integers with all $1 \leq V[i] \leq K$. Give (psuedocode is fine) the algorithm COUNTINGSORT that ends with $V$ in increasing order. (You may, and should, create auxiliary arrays.) Analyze the running time of COUNTINGSORT when $K = N$. (Note: It is not sufficient simply to give the answer, an analysis is called for.)

5. (20) For the following algorithms let $T(N)$ denote the total number of times the step after the WHILE step is reached. For the first algorithm give an exact formula for $T(N)$. For the second algorithm first give $T(N)$ as a precise sum. Then find $T(N)$ is the form $T(N) = \Theta(g(N))$ for a standard $g(N)$. Reasons please!

(a) $V=1$
    WHILE $V < N$
        do $V=2*V$
    END WHILE

(b) FOR $J=1$ TO $N$
    $W=J$
    WHILE $W \leq N$
        do $W=2*W$
    END WHILE
    END FOR

6. (20) Let $A[1 \cdots N]$ and $B[1 \cdots N]$ be arrays of numbers that are already in increasing order. Give an efficient algorithm for creating an array $TURING[1 \cdots 2N]$ consisting of the $2N$ entries in $A$ and $B$, placed in increasing order. How long (give a short reason) does your algorithm take?