**Theoretical Computer Science**
**Qualifying Exam**
**Fall 1996**

**Instructions.** Please work any 3 of the 5 problems given below.

**Problem 1.** In discussing the average behavior of quicksort, the simplifying assumption is often made that a list of size $n$ splits into two lists of size exactly $n/2$. This leads to the recurrence relation

$$T(n) = n + 2 * T(n/2),$$

which implies

$$T(n) \approx n \log_2 n = 1.442...n \ln n.$$

However, the actual average time for quicksort is known to be

$$\approx 2n \ln n.$$

For this problem you are to consider some modifications to the simple model that may give better predictions.

a) In assessing the typical behavior of quicksort, it would be better to assume that the subproblems have sizes $a * n$ and $(1 - a) * n$, where

$$a = lim_{n \to \infty} \frac{\text{expected size of the largest subproblem}}{n}$$

Compute $a$. What value of $T(n)$ results?

b) Suppose all subproblems had sizes $a * n$ and $(1 - a) * n$ for some fixed value of $a$. What value of $a$ makes $T(n) \approx 2n \ln n$?

Comment: Your answers will involve various numerical constants. It's enough to determine them implicitly, if you can't do the numerical calculations.

Hint: It is enough to consider leading terms of the running time functions.

**Problem 2.** Savitch's theorem is the following:

**Theorem.** If L is in NSPACE $(S(n))$, then L is in DSPACE$(S^2(n))$ provided that $S(n)$ is fully space constructible and $S(n) \geq \log_2 n$.

a) Give a proof of this theorem.

b) Explain why $S(n)$ is required to be fully space constructible and $\geq \log_2 n$.

**Problem 3.** For this problem you are to consider programs that have $n$ instructions, each of the following type:

$$L_i : \text{Flip a fair coin. If heads, go to } L_j \text{ , if tails, to } L_k.$$

Assume that the program starts at $L_1$, and stops when $L_n$ is reached. The goal is to design an "alarm clock" which "goes off" as late as is possible.

a) Show how to achieve expected termination time $\Theta(2^n)$. This gives an alarm clock using $\log n$ bits to measure time up to $2^n$.

b) Give the best upper bound you can for the expected termination time of an alarm clock with $n$ states.

**Problem 4.** Let $w$ be a word in $\Sigma^*$ of length $n$. Let

$$L_w = \{ \text{ x : there are words u,v with x = uwv } \}.$$

a) Show that any deterministic finite automaton recognizing $L_w$ has $\geq n + 1$ states.

b) Exhibit such an automation with exactly $n + 1$ states.

c) What is the relevance of this result to the Knuth/Morris/Pratt pattern matching algorithm?

**Problem 5.** The longest common subsequence problem is often solved using a dynamic programming algorithm. Given two sequences of lengths $m$ and $n$, the longest common subsequence is computed in time and space $\Theta(mn)$.

Example. Given sequences ABCBDAB and BDCABA, BCBA is a longest common subsequence.

a) The dynamic programming algorithm is usually formulated so that a matrix $C$ is computed. Entry $c_{i,j}$ contains the length of the longest common subsequence that can be formed from the first $i$ characters of sequence 1 and the first $j$ characters of sequence 2. Thus, entry $c_{m,n}$ contains the length of the longest common subsequence that can be formed from the first $m$ characters of sequence 1 and the first $n$ characters of sequence 2 – that is, it is the length the longest common subsequence. Once the matrix $C$ has been computed we can reconstruct the longest common subsequence in $O(m + n)$ time.

a) Sketch an algorithm to reconstruct the longest common subsequence.

b) The dynamic programming algorithm discussed above uses $\Theta(mn)$ space. Describe an algorithm to compute the length of the longest common subsequence that uses only $\Theta(m + n)$ space.