Foundations of Computing
Depth Examination
Monday September 20, 1993
3:00 – 7:00 PM

Directions: Answer any FOUR of the seven questions below.

1. Consider the following (k,l)-*pruning game*. We are given k items $a_1, a_2, \ldots, a_k$. In each *round* of the game, at least half of the items take a *hit*. After an item has taken l hits, it becomes inactive. When an item becomes inactive, it no longer participates in the game.

   (a) Estimate the number of rounds, in the worst case, that are required to make at least $\frac{k}{2}$ of the items inactive.

   (b) Suppose n_i items are active in the i^{th} round. Estimate $\sum_{i=0}^{\infty} n_i$, again assuming that the hits are taken so that items remain active as long as possible.

2. Describe an efficient algorithm that, given a set of integers $S = \{x_1, x_2, \ldots, x_n\}$, finds a nonempty subset R of S, such that $\sum_{x_i \in R} x_i^3 = 0 \bmod n$.

3. The following problem arises in profiling programs. The Uniconnected Subgraph Problem is: given a directed graph, what is the minimum number of edges that must be removed so that the graph is uniconnected? A directed graph is uniconnected if, given any two vertices i and j, there is at most one (directed) path from i to j. Show that this problem is NP-complete.

   Suggestion: reduce the vertex cover problem to this problem.

4. Show that there is some constant c such that, for any recursively enumerable language, there is a 1-tape Turing machine which recognizes this language (i.e. the Turing-machine halts if and only if the input is in the language) with only c states.

5. The Shortest Path Problem (SSP) for undirected graphs is: given an undirected graph, two distinguished nodes a and b and a bound k, is the length of the shortest path between a and b at most k? Prove that SSP is complete for the class NLOG = NSPACE($\log n$), with respect to log-space computable reductions.

6. Let $x \geq 0$ be an integer. The following algorithm will compute S, the number of 1's in the binary representation of x:

```
S := 0
while (x != 0) do begin
        S := S+1
        x := x AND (x - 1)
        end
```

   (In this algorithm, AND is the bitwise "and" operation.)

   (a) Explain why the algorithm correctly computes S.

(b) What is the average number of loop iterations executed by this algorithm, if x is a random n-bit number?

(c) What is the variance of this quantity?

7. Describe an efficient time algorithm for the following problem. Given two words w_1 and w_2 over a finite alphabet, find the longest common subword. A common subword is a word that appears in both w_1 and w_2. What is the running time of your algorithm?