CS130N Problem set 1: Induction, recurrences and counting

January 4, 2001

(For problems 1 - 12 gaalis > suban; for problems 13 - 17 gaalis > rohitk)

1. Prove that all all natural numbers of the form $n^3 + 2n$ are divisible by 3.

2. Suppose we have coins of two different denominations, 3 paise and 5 paise. Show that it is possible to make up exactly any postage of 8 paise or more using stamps of these two denominations. In other words show that every positive integer $n \geq 8$ is expressible as $n = 3i + 5j$ where $i, j \geq 0$.

3. Let $F_0 = 0$, $F_1 = 1$, $F_2 = 1, \ldots$ be the Fibonacci sequence where for all $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. Let $\phi = (1 + \sqrt{5})/2$. Show that $F_n \leq \phi^{n-1}$ for all positive n.

4. Prove the following statement using **PMI**: If a line of unit length is given, then a line of length $\sqrt{n}$ can be constructed using *ruler and compass* for every positive integer n.

5. Prove by **PMI**, that every integer $n > 1$ is either a *prime* or a *product of primes.*

6. Consider the following *tiling* problem. You are given a $m \times m$ board with m squares in each row and m squares in each column where m is a power of 2. One arbitrary square on the board is distinguished as *special.* You are also given a supply of tiles, each of which looks like a 2×2 board with one square removed (L shaped). Your puzzle is to cover the board with these tiles so that each square is covered exactly once, with the exception of the *special* square, whic is not covered at all. Such a covering is called a *tiling.* Show, using **PMI**, that the *tiling* problem can always be solved.

7. Consider the following algorithm:

   function $sq(n)$
     if $(n = 0)$ **then return** 0
     else return $2n + sq(n-1) - 1$

   Prove that the above algorithm computes n^2 for $n \in \mathbb{N}$. Estimate the number of steps required as a function of n.

8. Consider the following algorithm:

```
function power(x, n)
  if (n = 0) then return 1
  else if even(n) then return sqr(power(x, n div 2))
  else return x * sqr(power(x, n div 2))
```

Prove that the above algorithm computes x^n for all $x \in \mathbb{N}$ and all $n \in \mathbb{N}$. Also show that the number of steps required is bounded by $\lceil 2 \lg n \rceil$.

9. Consider the following algorithm for computing the *maximum* and *minimum* element in an array S of size $n \geq 2$. Assume that n is a perfect power of two.

```
function MAXMIN(S)
  if (size(S) = 2) then
   begin
    let S = [a, b]
    return (max(a, b), min(a, b))
   end
  else
   begin
    split S into two sub-arrays S1 and S2 each with half the elements;
    (max1, min1) ← MAXMIN(S1);
    (max2, min2) ← MAXMIN(S2);
    return (max(max1, max2), min(min1, min2));
   end
```

Let $T(n)$ be the total number of comparisons required to compute the above function. Clearly, $T(2) = 1$, and to solve a problem of size n we have to solve two sub-problems of size $n/2$ and carry out two additional comparisons. Thus, we have the following recurrence for $T(n)$:

$$\begin{aligned} T(2) &= 1 \\ T(n) &= 2T(n/2) + 2 \text{ for } n > 2 \end{aligned}$$

Show that at least $(3n/2) - 2$ comparisons are necessary.

10. Solve the following recurrences. Try to obtain tight asymptotic upper and lower bound in each case. (Can the tutors please explain O, Ω, Θ, o, ω, ...?). Assume, in each case, $T(n)$ is constant for $n \leq 2$.

   (a) $T(n) = T(\lceil n/2 \rceil) + 1$
   (b) $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1$
   (c) $T(n) = T(\lfloor n/2 \rfloor) + n$
   (d) $T(n) = 2T(\lfloor n/2 \rfloor) + n$
   (e) $T(n) = 3T(\lfloor n/2 \rfloor) + n$

(f) $T(n) = 4T(\lfloor n/2 \rfloor) + n$

(g) $T(n) = T(\lfloor \sqrt{n} \rfloor) + n$

11. Find the number of ways for a rook to move from the southwest corner of a $p \times q$ chessboard to the northeast corner by moving one square at a time eastward or northward only. Note that rook is a chesspiece that can move horizontally and vertically on a chessboard.

12. Suppose you have an infinite supply of coins of denomination $50p$, $25p$, $10p$, $5p$ and $1p$. In how many ways can you generate change for a given amount, say Rs. 1/- $= 100p$. Given that a function $d(n)$ is available which gives the denomination of the n^{th} type of coin, develop a recursive algorithm to count the number of ways to generate change for a given amount. What can you say about the number of steps required for the computation?

13. In to how many regions is the plane divided by n straight lines in general position (no two parallel, no three concurrent)? How many of the regions are bounded?

14. Suppose there are n points on a circle. We draw all $\binom{n}{2}$ chords between those points. Suppose that no three of these chords pass through a single point. How many regions do these chords divide the interior of the circle in to ?

15. In a sufficiently large set, n subsets are chosen. What is the maximum number of sets that can be formed from these n sets using (for example) intersection and complementation? What does "sufficiently large" mean?

16. If every two cities are joined by a one-way road, then is it possible to find a starting city C and a route from C that passes through every city exactly once?

17. In an ancient village, there were some green-eyed and blue-eyed persons. One fine day, God instructed them, "the day on which you come to know that you are a green-eyed, you should commit suicide ..." He also assured them that there was at least one green-eyed among them. Well, all the villagers were very intelligent and strict followers of God. But, no one knew colour of his own eyes! They didn't have mirrors. They couldn't even communicate with each other. All that they could do is to see colour of other's eyes. It happened that on 20^{th} day, all the green-eyed people committed suicide. So, how many green-eyed were there ?