Problem Set 7

Parvesh Adi Lachman

November 2023

1 Problem 1

Consider the algorithm below, which takes an $n \geq 0$ and finds it remainder when divided by $c \geq 1$

function REMAINDER(n):
  if $n \leq c - 1$ **then**
    return n
  else
    return Remainder($n - c$)
  end if
end function

Claim: Let $c \geq 1$. For any $n \geq 0$, **remainder(n)** $= n \mod c$.

2 Problem 2

Claim: Let $n, c \geq 1$ and $c \leq n$. The number of simple paths of length c in the complete graph on n nodes is $\frac{n!}{(n-c-1)!}$ which is equal to $n(n-1)\cdots(n-c)$.

complete graph K_n: an undirected graph on n nodes with an edge between every pair of nodes.

simple path: a sequence of distinct nodes with edges between consecutive nodes in the sequence.

length of a path: the number of *edges* in the path (**not** number of nodes).

Proof: Let G be a complete graph on n nodes. Let $v_1, v_2, \ldots, v_n$ be the nodes of G. Let P be a simple path of length c in G. Then P is a sequence of c distinct nodes in G. Since P is a simple path, the nodes in P are distinct. Thus, P is a permutation of c distinct nodes in G. Since G has n nodes, there are n choices for the first node in P, $n-1$ choices for the second node in P, and so on. Thus, there are $n(n-1)\cdots(n-c)$ simple paths of length c in G.

3 Problem 3

Recall the Fibonacci numbers, as defined by:

$$f_1 = 1$$
$$f_2 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for } n \geq 3$$

Recall the Sharp numbers from PS6, as defined by:

$$s_1 = 2$$
$$s_2 = 4$$
$$s_n = s_{n-1} + s_{n-2} \text{ for } n \geq 3$$

Claim: For all $n \geq 3$, $s_n = 4 \cdot f_{n-1} + 2 \cdot f_{n-2}$.