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 $n \geq 0$, $\textbf{remainder}(n) = n \mod c$.

Step 0: For $n \geq 0$, we want to show that $\textbf{remainder}(n) = n \mod c$.

Step 1: For any $n \geq 0$, Let $P(n)$ be the property that $\textbf{remainder}(n) = n \mod c$.

Step 2: As base cases, consider when

$n = 0$. We will show that $P(0)$ is true: that is $\textbf{remainder}(0) = 0 \mod c$. Fortunately, this is true since $c \geq 1$ and in the algorithm, if $n \leq c - 1$, and in this case $n = 0$, $c \geq 1$. This implies $0 \leq 1 - 1 = 0 \leq 0$ which is true, then $\textbf{remainder}(n) = n$. Thus, $\textbf{remainder}(0) = 0$, so RHS $= 0$. Also, $0 \mod c = 0$, so LHS $= 0$. Thus, LHS = RHS, so $P(0)$ is true.

$n = 1$. We will show that $P(1)$ is true: that is $\textbf{remainder}(1) = 1 \mod c$. Fortunately, this is true since $c \geq 1$ and in the algorithm, if $n \leq c - 1$, then $\textbf{remainder}(n) = n$. Thus, $\textbf{remainder}(1) = 1$, so RHS $= 1$. Also, $1 \mod c = 1$, so LHS $= 1$. Thus, LHS = RHS, so $P(1)$ is true.

Step 3: Let $k \geq 2$. For the induction hypothesis, suppose that $P(0), P(1), \ldots, P(k)$ are true, or equivalently, that for all $0 \leq k' \leq k : P(k')$. That is, suppose that $\textbf{remainder(}k'\textbf{)} = k' \mod c$.

Step 4: Now we prove that $P(k+1)$ is true, using our induction assumptions that $P(0), P(1), \ldots, P(k)$ are true. That is, we prove that $\textbf{remainder(}k+1\textbf{)} = (k+1) \mod c$.

Step 5: The proof that $P(k+1)$ is true (given that $P(0), P(1), \ldots, P(k)$ are true) is as follows:

$$\begin{array}{rcll}
\text{Left hand side of } P(k+1) & = & \textbf{remainder(}k+1\textbf{)} & \\
 & = & \textbf{remainder(}(k+1)-c\textbf{)} & \text{By def of algorithm, since } k+1 \geq c-1 \\
 & = & ((k+1)-c) \mod c & \text{By IH, since } 0 \leq (k+1)-c \leq k \\
 & = & (k+1) \mod c - c \mod c & \text{By def of mod} \\
 & = & (k+1) \mod c & \text{Since } c \geq 1,\ c \mod c = 0 \\
 & = & \text{Right hand side of } P(k+1) &
\end{array}$$

Step 6: The steps above have shown that for any $k \geq 2$, if $P(0), P(1), \ldots, P(k)$ are true, then $P(k+1)$ is also true. Combined with the base cases, which show that $P(0)$ and $P(1)$ are true, we have shown that for all $n \geq 0$, $P(n)$ is true, as desired.

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).

3 Problem 3

Recall the Fibonacci numbers, as defined by:

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

Recall the Sharp numbers from PS6, as defined by:

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

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