

1 Inductive Proofs

Prove each of the following claims by induction

Claim 1. *The sum of the first n even numbers is $n^2 + n$. That is, $\sum_{i=1}^n (2i) = n^2 + n$*

Proof. We will prove the claim by induction.

1. Base Case. The sum of the first 1 even integer is 2. $\sum_{i=1}^1 = 1^2 + 1 = 2$, as desired.
2. Induction Hypothesis. Suppose for all $m < n$, the sum of the first m integers is $m^2 + m$.
3. Induction Step. Want to show that $\sum_{i=1}^n (2i) = n^2 + n$.

$$\begin{aligned}\sum_{i=1}^n (2i) &= \sum_{i=1}^{(n-1)} (2i) + 2n, \text{ by IH} \\ &= (n-1)^2 + (n-1) + 2n \\ &= n^2 - 2n + 1 + n - 1 + 2n \\ &= n^2 + n,\end{aligned}$$

as desired.

□

Claim 2. $\sum_{i=1}^n \frac{4}{5^i} = 1 - \frac{1}{5^n}$

Proof. We will prove the claim by induction.

1. Base Case, $n = 1$. $\sum_{i=1}^1 = \frac{4}{5^1} = \frac{4}{5} = 1 - \frac{1}{5^1}$, as desired.
2. Induction Hypothesis. Suppose for all $m < n$, the sum of the first m integers is $1 - \frac{1}{5^m}$.
3. Induction Step.

$$\begin{aligned}\sum_{i=1}^n \frac{4}{5^i} &= \sum_{i=1}^{(n-1)} \frac{4}{5^i} + \frac{4}{5^n}, \text{ by IH} \\ &= 1 - \frac{1}{5^{n-1}} + \frac{4}{5^n} \\ &= 1 + \left(-\frac{5}{5^n} + \frac{4}{5^n}\right) \\ &= 1 - \frac{1}{5^n},\end{aligned}$$

as desired.

□

Claim 3. For every $n \geq 1$, $4^n - 1$ is divisible by 3. In other words, for every $n \geq 1$, there exists some integer constant k_n such that $4^n - 1 = 3k_n$. Note that each power of 4 has a different k .

Proof. We will prove the claim by induction.

1. Base Case, $n = 1$. $4^1 - 1 = 4 - 1 = 3 = 3 * 1$. So $k_n = 1$ in this case, as desired.
2. Induction Hypothesis. Suppose for all $m < n$, $4^m - 1 = 3 * k_m$ for some integer k_m .
3. Induction Step.

$$\begin{aligned}
 4^n - 1 &= 4 \cdot 4^{n-1} - 1 \\
 &= 3 \cdot 4^{n-1} + (4^{n-1} - 1) \text{ by IH} \\
 &= 3 \cdot 4^{n-1} + 3 \cdot k_m, \text{ for some integer } k_m \\
 &= 3 \cdot (4^{n-1} + k_m),
 \end{aligned}$$

where $(4^{n-1} + k_m)$ is the desired integer k_n . Therefore $4^n - 1$ is divisible by 3.

□

2 Recursive Invariants

The function `minPos`, given below in pseudocode, takes as input an array A of size n numbers and returns the smallest positive number in the array. If no positive numbers appear in the array, it returns $+\infty$. (Note that zero is neither positive nor negative.) Using induction, prove that the `minPos` function works correctly. Clearly state your recursive invariant at the beginning of your proof.

```

Function minPos(A,n)
  If n = 0 Then
    Return +Infinity
  Else
    best <- minPos(A,n-1)
    If A[n-1] < best And A[n-1] > 0 Then
      best <- A[n-1]
    EndIf
    Return best
  EndIf
EndFunction

```

Invariant: The function minPos will always return the smallest number in a given array A of size n , unless there is no positive number in the array in which case the function will return $+\infty$. **Claim:** The invariant above holds for all $n \geq 0$.

Proof. We will prove the claim by induction. (Note: ∞ isn't a number so it can't be an entry of A .)

1. Base cases, $n = 0, 1$. When $n = 0$, the function enters the 'if' statement and returns $+\infty$. $+\infty$ is the desired answer because with an empty array, there are no positive numbers in the array. When $n = 1$, the function enters the 'else' statement. If the only entry in A is non-positive, the function recursively calls minPos with inputs $A, n - 1 = 0$, and sets 'best' to $+\infty$. Since $A[n - 1] = A[1 - 1] = A[0] < \infty = \text{'best'}$, the function does not enter the 'if' sub-loop and returns current value of 'best' which is $+\infty$, as desired. If $n = 1$ and the only entry in A is positive, 'best' is set to $\text{minPos}(A, n - 1) = \text{minPos}(A, 1 - 1 = 0) = +\infty$. The sub-if loop evaluates ' $A[n-1] \leq \text{best}$ ' to be true and ' $A[n-1] \leq 0$ ', so 'best' is set to $A[n - 1] = A[0]$ which is the only entry in the array. The current value of 'best' is then returned. And since it is the only positive number in A , this is the desired answer to be returned.
2. Induction hypothesis: assume that for all $m < k$ and k is a natural number, the invariant holds when $n = m$.
3. Induction step: want to show that the invariant holds for $n = k$. There are three cases that we have to prove: 1. when no entry of A is positive, in which case the desired answer is $+\infty$; 2. the smallest positive number is in the first $n - 1$ entries of A ; 3. the smallest positive number is the n^{th} entry of A .

Let $k > 1$ (not a base case), then we enter the 'else' statement of the function.

Consider case 1, where the desired answer is $+\infty$. The current value of 'best' is $\text{minPos}(A, n - 1)$; by induction hypothesis, $\text{minPos}(A, n - 1)$ returns the desired answer which is $+\infty$ since no entries of A are positive. We go onto the 'if' statement: $A[n - 1] < \text{best}$ is true because $A[n - 1] < 0 < +\infty = \text{'best'}$; $A[n - 1] > 0$ is false because $A[n - 1] < 0$. Therefore we do not enter the inside of the if statement. 'best' remains unchanged and returned. Therefore the returned answer is $+\infty$ as desired.

For case 2, by induction hypothesis 'best' is set to $\text{minPos}(A, n - 1)$ which returns the smallest positive number in the first $n - 1$ entries of A by IH. The current value of 'best' is also the smallest positive number in the entire array by assumption. Let us look at the if statement: If $A[n - 1]$ is positive, it is still greater than current value of 'best' by assumption, so we do not enter the if loop. If $A[n - 1]$ is not positive, ' $A[n - 1] > 0$ ' is evaluated to be false, so we do not enter the if loop. Therefore under current assumptions,

we never enter the if loop, and ‘best’ remains unchanged and eventually returned, as desired.

For case 3, where the smallest positive number is the n^{th} entry of A , immediately after we enter the else statement, the current value of ‘best’ is not the smallest positive number in the array by IH. We want to show that the two conditionals of the if statement stays true in all possible circumstances under current assumption. ‘ $A[n - 1] < \text{best}$ ’ stays true because the current value of ‘best’ is the output of $\text{minPos}(A, n - 1)$ which is either $+\infty$ or a positive number greater than $A[n - 1]$ by IH. ‘ $A[n - 1] > 0$ ’ stays true because we assumed that $A[n-1]$ is a positive number. Therefore we always enter the if loop, and ‘best’ is set to the n^{th} entry of A and returned, as desired.

□