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,* $\sum_{i=1}^{n} 2i = n^2+n$

Base Case: for $n = 1$
$\sum_{i=1}^{1} 2i = 2(1) = 2$
$n^2 + n = 1^2 + 1 = 2$
$2 = 2$
Statement is true for $n = 1$.

Inductive hypothesis:
Assume that $\sum_{i=1}^{n} 2i = n^2 + n$ for all n such that $1 \leq n \leq k$.

Inductive step:
Show that $\sum_{i=1}^{k+1} 2i = (k+1)^2 + (k+1) = k^2 + 2k + 1 + k + 1 = k^2 + 3k + 2$.

$$\begin{aligned}
\sum_{i=1}^{k+1} 2i &= \sum_{i=1}^{k} 2i + 2(k+1) \\
&= (k^2 + k) + 2k + 2 && \text{by inductive hypothesis} \\
&= k^2 + 3k + 2
\end{aligned}$$

Thus, $\sum_{i=1}^{n} 2i = n^2 + n$ for all n.

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

Base Case: for $n = 1$
$\sum_{i=1}^{1} \frac{1}{2^i} = \frac{1}{2^1} = \frac{1}{2}$
$1 - \frac{1}{2^1} = \frac{1}{2}$
$\frac{1}{2} = \frac{1}{2}$
Statement is true for $n = 1$.

Inductive hypothesis:
Assume that $\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}$ for all n such that $1 \leq n \leq k$.

Inductive step:
Show that $\sum_{i=1}^{k+1} \frac{1}{2^i} = 1 - \frac{1}{2^{k+1}}$.

$$\begin{aligned}
\sum_{i=1}^{k+1} \frac{1}{2^i} &= \sum_{i=1}^{k} \frac{1}{2^i} + \frac{1}{2^{k+1}} \\
&= (1 - \frac{1}{2^k}) + \frac{1}{2^{k+1}} && \text{by inductive hypothesis} \\
&= 1 - \frac{1 * 2}{2^k * 2^1} + \frac{1}{2^{k+1}} \\
&= 1 + \frac{-2 + 1}{2^{k+1}} \\
&= 1 - \frac{1}{2^{k+1}}
\end{aligned}$$

Thus, $\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}$ for all n.

Claim 3. $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

Base Case: for $n = 0$

$\sum_{i=0}^{0} 2^i = 2^0 = 1$

$2^{n+1} - 1 = 2^{0+1} - 1 = 1$

$1 = 1$

Statement is true for $n = 0$.

Inductive hypothesis:

Assume that $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ for all n such that $0 \leq n \leq k$.

Inductive step:

Show that $\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$.

$$\begin{aligned}
\sum_{i=0}^{k+1} 2^i &= \sum_{i=0}^{k} 2^i + 2^{k+1} \\
&= (2^{k+1} - 1) + 2^{k+1} && \text{by inductive hypothesis} \\
&= 2 * 2^{k+1} - 1 \\
&= 2^{k+2} - 1
\end{aligned}$$

Thus, $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ for all n.

2 Recursive Invariants

The function `minEven`, given below in pseudocode, takes as input an array A of size n of numbers. It returns the smallest *even* number in the array. If no even numbers appear in the array, it returns positive infinity ($+\infty$). Using induction, prove that the `minEven` function works correctly. Clearly state your recursive invariant at the beginning of your proof.

```
Function minEven(A,n)
  If n = 0 Then
    Return +∞
  Else
    Set best To minEven(A,n-1)
    If A[n-1] < best And A[n-1] is even Then
      Set best To A[n-1]
    EndIf
    Return best
  EndIf
EndFunction
```

Recursive invariant: the function returns the smallest even number in the array, up to the nth value.

Base Case: for $n = 0$

If the array is empty, it should return $+\infty$. In the code, the size of the array is defined as n, and if $n = 0$, it returns $+\infty$. Thus, the function works for the base case.

Inductive hypothesis:

Assume that the function returns the smallest even number for arrays of size n such that $0 \leq n \leq k$.

Inductive step:

Show that `minEven(A, k+1)` returns the smallest even number in the array A up to the (k+1)th value.

`best` will be set to `minEven(A, k)`, which (we know from our inductive hypothesis) will return the smallest even number up to the kth value. Then if the (k+1)th value is even and less than `best`, it will become the new value of `best`. If it is odd or greater than `best`, `best` will remain the same.

This function works because it recursively calls itself on the entire array except for the last element. This means that we are comparing the smallest even number from the first $n - 1$ elements of the array to the last element of the array to see if the last element is actually the smallest even number in the array.

Thus, the function `minEven` will work for all n.