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

1. Base case: $i = 1$
LHS: $2*(1) = 2$
RHS: $(1)^2 + 1 = 2$
Thus, LHS = RHS.

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

3. Inductive Step:

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

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

Thus, we have shown that $\sum_{i=1}^{k+1} 2i = (k+1)^2 + (k+1)$.

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

1. Base case: $i = 1$
LHS: $\frac{1}{2^(1)} = \frac{1}{2}$
RHS: $1 - \frac{1}{2^(1)} = 1 - \frac{1}{2} = \frac{1}{2}$
Thus, LHS = RHS.

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

3. Inductive Step:

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

$$\begin{aligned}
\sum_{i=1}^{k+1} \tfrac{1}{2^i} &= \sum_{i=1}^{k} \tfrac{1}{2^i} + \tfrac{1}{2^{(k+1)}} \\
&= 1 - \tfrac{1}{2^k} + \tfrac{1}{2^k} * \tfrac{1}{2} \\
&= 1 - \tfrac{1}{2} * \tfrac{1}{2^k} \\
&= 1 - \tfrac{1}{2^{(k+1)}}
\end{aligned}$$

Thus, we have shown that $\sum_{i=1}^{k+1} \frac{1}{2^i} = 1 - \frac{1}{2^{(k+1)}}$.

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

1. Base case: $i = 0$
LHS: $2^0 = 1$
RHS: $2^{(0+1)} - 1 = 2 - 1 = 1$
Thus, LHS = RHS.

2. Inductive Hypothesis:
Assume that $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ for all $0 \leqslant n \leqslant k$.

3. Inductive Step:

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

$$\begin{aligned}
\sum_{i=0}^{k+1} 2^i &= \sum_{i=0}^{k} 2^i + 2^{(k+1)} \\
&= 2^{(k+1)} - 1 + 2^{(k+1)} \\
&= 2^k * (2 + 2) - 1 \\
&= 2^k * 4 - 1 \\
&= 2^k * 2^2 - 1 \\
&= 2^{(k+2)} - 1
\end{aligned}$$

Thus, we have shown that $\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1$.

2 Recursive Invariants

The function `minEven`, given below in pseudocode, takes as input an array A of size n of numbers. It returns the smallestest *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 is 0 Then
    Return +infinity
  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:

$P(n)$ = if there is even number in the array A, minEven() will return the smallest even number in the array. Otherwise (if there is no even number in the array A), minEven() will return positive infinity ($+\infty$).

1. Base case: $n = 0$

If $n = 0$, it means the array A is empty, suggesting that there is no even number in the array A, and the algorithm will return $+\infty$.

2. Inductive Hypothesis:

Assume that the algorithm minEven() works correctly for all arrays of size n where $0 \leqslant n \leqslant k$.

3. Inductive Step:

Show that the algorithm minEven() works correctly for arrays with $k + 1$ elements.

minEven$(A, k + 1)$:

If $k + 1 > 0$, minEven$(A, k + 1)$ will call minEven(A, k), which by our Inductive Hypothesis works correctly. Thus, minEven$(A, k + 1)$ will return the current smallest even number in the array A.

If the number at index k is smaller than the current smallest even number returned by minEven(A, k), which means the value at index k is now the smallest even number in the array, the current smallest even number (stored in the variable *best*) will be updated to the value at index k.

Otherwise, the current smallest even value in the array A will be returned.