1 Inductive Proofs

Solve the following problems with induction

1. The sum of the first n even numbers is $n^2 + n$. That is,

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

proof.

Base case (n = 1):

$$\text{LHS: } \sum_{i=1}^{1} 2i = 2$$
$$\text{RHS: } 1^2 + 1 = 2$$

Since LHS = RHS, the base case holds true

Inductive Hypothesis: Assume that

$$\sum_{i=1}^{n} 2i = n^2 + n \text{ for all n such that } 1 \leq n \leq k$$

Inductive Step:

Based on the inductive hypothesis assumption, I must show $\displaystyle\sum_{i=1}^{k+1} 2i = (k+1)^2 + (k+1) = k^2 + 3k + 2$

If we expand the last term, we get:

$$\sum_{i=1}^{k+1} 2i = 2(k+1) + \sum_{i=1}^{k} 2i$$

With our inductive hypothesis, we have:

$$= 2(k+1) + k^2 + k$$
$$= k^2 + k + 2k + 1$$
$$= k^2 + 3k + 2$$

Therefore, the inductive step holds true.

Conclusion: Since the inductive step and the base case are true, by induction $\sum_{i=1}^{n} 2i = n^2 + n$ must be true for any value of $n \geq 1$.

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

proof.

Base case (n = 1):

$$\text{LHS: } \sum_{i=1}^{1} \frac{1}{2^i} = \frac{1}{2}$$
$$\text{RHS: } 1 - \frac{1}{2^1} = \frac{1}{2}$$

Since LHS = RHS, the base case holds true

Inductive Hypothesis: Assume that

$$\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n} \text{ for all n such that } 1 \le n \le k$$

Inductive Step:

Based on the inductive hypothesis assumption, I must show that $\sum_{i=1}^{k+1} \frac{1}{2^i} = 1 - \frac{1}{2^{k+1}}$

If we expand the last term, we get:

$$\sum_{i=1}^{k+1} \frac{1}{2^i} = \frac{1}{2^{k+1}} + \sum_{i=1}^{k} \frac{1}{2^i}$$

With our inductive hypothesis, we have:

$$= \frac{1}{2^{k+1}} + 1 - \frac{1}{2^k}$$

Finding common denominator (with expo law $2^k * 2^m = 2^{k+m}$):

$$= \frac{1}{2^{k+1}} + 1 - \frac{1}{2^k} * \left(\frac{2^1}{2^1}\right)$$

$$= \frac{1}{2^{k+1}} + 1 - \frac{2}{2^{k+1}}$$

$$= 1 - \frac{1}{2^{k+1}}$$

Therefore, the inductive step holds true.

Conclusion: Since the inductive step and the base case are true, by induction $\sum_{i=1}^{n} \frac{1}{2^i} = 1 - \frac{1}{2^n}$ must be true for any value of $n \ge 1$.

3.

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

proof.

Base case (n = 0):

$$\text{LHS: } \sum_{i=0}^{0} 2^i = 1$$

$$\text{RHS: } 2^{0+1} - 1 = 2 - 1 = 1$$

Since LHS = RHS, the base case holds true

Inductive Hypothesis: Assume that

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

Inductive Step:

Based on the inductive hypothesis assumption, I must show that $\sum_{i=0}^{k+1} 2^i = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$

If we expand the last term, we get:

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

With our inductive hypothesis, we have:

$$= 2^{k+1} + 2^{k+1} - 1$$

$$= k^2 + k + 2k + 1$$

$$= 2 * (2^{k+1}) - 1$$

2 is same as 2^1 (to apply law $2^k * 2^m = 2^{k+m}$)

$$= 2^1 * (2^{k+1}) - 1$$

$$= 2^{(k+1+1)} - 1 = 2^{k+2} - 1$$

Therefore, the inductive step holds true.

Conclusion: Since the inductive step and the base case are true, by induction $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ must be true for any value of $n \geq 0$.

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:
Let $P(n)$ be the function that represents minEven such that $P(n)$ either returns the lowest even number in the first n elements of an array or returns positive infinity if no even numbers are within the first n elements.

Base Case (n = 0)

1. For $P(0)$, the function has an input of the first 0 elements in an array, or in other words no elements at all. Since an empty array can never contain an even number, it must be true that $P(0)$ outputs positive infinity.
2. The first if statement in minEven would check if n is equal to zero. Since n is equal to zero, it returns positive infinity.
3. Therefore, the base case of n = 0 follows the recursive invariant and thus holds true.

Induction Hypothesis
Assume that $P(n)$ functions properly for all inputs of size $0 \leq n \leq k$ such that $P(n)$ either returns the lowest even number in the first n terms of an array or that it returns positive infinity if the array only contains odd numbers in the first n elements.

Inductive Step
Goal: Show that P(K+1) always outputs the proper value.

1. The initial call of the function will be minEven(A, K+1) where A is an array of at least K+1 elements.
2. Since K+1 is greater than zero, the first if statement in minEven is false, causing the function to enter the else statement

3. In the else statement, minEven(A, K+1) calls minEven(A, K) and sets it to the variable **best**.
    (a) Based on the Inductive Hypothesis, it can be assumed that minEven(A, K) works properly and either returns the lowest even number in the first K elements or returns positive infinity if there are no even numbers.
4. In the second if statement, if A[K] is a smaller even number than **best**, then **best** is set to A[K].
    (a) Based on the previous step, it can be assumed that at this point **best** is either the smallest even number from the first K elements or positive infinity if the first K elements of the array does not contain an even number.
    (b) If A[K] is a smaller even number than **best**, that means that A[K] must be the smallest even number in the first K+1 elements of the array. In this case, **best** will be set to A[K]
    (c) If A[K] is not a smaller even number than **best**, that means that the current value of **best** is either the smallest even number in the first K+1 terms of the array or is positive infinty if there are no even numbers in the first K+1 elements. In this case, best retains its value.
    (d) In both situations, the value stored in **best** is either the lowest even number in the first K+1 elements of the array, or positive infinity if the array does not contain any even numbers in the first K+1 elements.
5. minEven(A,K+1) then returns **best**.
    (a) As stated in the previous step, **best** represents either the smallest even number in the first K+1 elements of the array or positive infinity if there were no even numbers in the first K+1 elements. Since **best** is equal to the value that minEven is intended to output, it must be true that minEven(A, K+1) always returns the correct value, and therefore P(K+1) works as intended.
6. Since P(K+1) always outputs the intended results, the inductive step is proven to be true

Conclusion: Since the inductive step and the base case are true, by induction P(n) is true for all $n \geq 0$, and therefore minEven works correctly.