Exercise Class 9 – Solutions
Proof by Induction

---

1. We want to prove that:

$$\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$$

For the **base case** ($n = 0$), the left-hand side is $0^2 = 0$ and the right-hand side is $(0 \cdot 1 \cdot 1)/6 = 0$, which are equal.

For the **inductive step** ($n > 0$), we will assume, as our **inductive hypothesis** that:

$$\sum_{i=0}^{k} i^2 = k(k+1)(2k+1)/6$$

and then prove that:

$$\sum_{i=0}^{k+1} i^2 = (k+1)(k+2)(2(k+1)+1)/6 \tag{1}$$

Starting from the left-hand side of equation (1):

$$\begin{aligned}
\sum_{i=0}^{k+1} i^2 &= \sum_{i=0}^{k} i^2 + (k+1)^2 \\
&= (k(k+1)(2k+1)/6) + (k+1)^2 && \text{by the inductive hypothesis} \\
&= (k(k+1)(2k+1) + 6(k+1)^2)/6 && \text{rearranging} \\
&= (2k^3 + 2k^2 + k^2 + k + 6k^2 + 12k + 6)/6 && \text{rearranging} \\
&= (2k^3 + 9k^2 + 13k + 6)/6 && \text{rearranging}
\end{aligned}$$

And from the right-hand side of equation (1):

$$\begin{aligned}
(k+1)(k+2)(2(k+1)+1)/6 &= (k^2 + 3k + 2)(2k+3)/6 \\
&= (2k^3 + 6k^2 + 4k + 3k^2 + 9k + 6)/6 \\
&= (2k^3 + 9k^2 + 13k + 6)/6
\end{aligned}$$

These are equal, which completes the proof for the inductive step.

Therefore:

$$\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$$

2. We want to prove that: `append l [] = l`.

   For the **base case** (`l = []`) the left-hand side is:

   `append l [] = append [] [] = []` (by the definition of `append`)

   and the right-hand side is:

   `l = []`

   which are equal. So the base case holds.

   Now, for the **inductive step** (`l = hd::tl`).

   We will assume, as our **inductive hypothesis** that:

   `append tl [] = tl`

   We need to prove that:

   `append hd::tl [] = hd::tl`.

   Starting from the left-hand side:

   ```
       append hd::tl []
    =  hd :: (append tl [])   by the definition of append
    =  hd :: tl               by the inductive hypothesis
   ```

   which proves the equality, as required.

   Therefore: `append l [] = l` for any list `l`.

3. We want to prove that: `mem x l = mem x (setify l)`.

   For the **base case** (`l = []`) the left-hand side is:

   `mem x l = mem x [] = false` (by the definition of `mem`)

   and the right-hand side is:

   `mem x (setify []) = mem x [] = false` (by the definitions of `setify` and `mem`)

   which are equal, so the base case holds.

   Now, for the **inductive step** (`l = hd::tl`).

   We will assume, as our **inductive hypothesis** that:

   `mem x tl = mem x (setify tl)`

   We need to prove that:

   `mem x hd:tl = mem x (setify hd:tl)`

   Starting from the right-hand side:

   ```
       mem x (setify hd::tl)
    =  mem x (if mem hd tl then setify tl else hd::(setify tl))
   ```
   (by the definition of `setify`)

   We split our proof into two cases to handle the `if`:

   Case 1 (`mem hd tl`)
   ```
    =  mem x (setify tl)
   ```
   (expanding the `if`)
   ```
    =  mem x tl
   ```
   (by the inductive hypothesis)
   ```
    =  (hd = x) || (mem x tl)
   ```
   (since `mem hd tl`)
   ```
    =  mem x hd::tl
   ```
   (by the definition of `mem`)

   Case 2 (`not mem hd tl`)
   ```
    =  mem x (hd::(setify tl))
   ```
   (expanding the `if`)
   ```
    =  (hd = x) || (mem x (setify tl))
   ```
   (by the definition of `mem`)
   ```
    =  (hd = x) || (mem x tl)
   ```
   (by the inductive hypothesis)
   ```
    =  mem x hd::tl
   ```
   (by the definition of `mem`)

   This proves the equality, as required.

   Therefore: `mem x l = mem x (setify l)` for any list `l`.

4. We want to prove that: `setify (setify l) = setify l`.

   For the **base case** (`l = []`) the left-hand side is:

   `setify (setify l) = setify (setify []) = setify [] = []` (by the definition of `setify`)

   and the right-hand side is:

   `setify [] = []` (again, by the definition of `setify`)

   which are equal, so the base case holds.

   Now, for the **inductive step** (`l = hd::tl`).

   We will assume, as our **inductive hypothesis** that:

   `setify (setify tl) = setify tl`

   We need to prove that:

   `setify (setify hd:tl) = setify hd:tl`

   Starting from the left-hand side:

```
    setify (setify hd::tl)
 =  setify (if mem hd tl then setify tl else hd::(setify tl))
```
   (by the definition of `setify`)

   We split our proof into two cases to handle the `if`:

   Case 1 (`mem hd tl`)
```
 =  setify (setify tl)
```
   (expanding the `if`)
```
 =  setify tl
```
   (by the inductive hypothesis)
```
 =  setify hd::tl
```
   (by the definition of `setify`, since `mem hd tl`)

   Case 2 (`not mem hd tl`)
```
 =  setify(hd::(setify tl)
```
   (expanding the `if`)
```
 =  if mem hd (setify tl) then setify (setify tl) else hd::(setify (setify tl))
```
   (by the definition of `setify`)
```
 =  if mem hd (setify tl) then setify tl else hd::(setify tl)
```
   (using the inductive hypothesis, twice)
```
 =  if mem hd tl then setify tl else hd::(setify tl)
```
   (using question 3)
```
 =  setify hd::tl
```
   (by the definition of `setify`)

   So the left- and right-hand sides are equal, as required.

   Therefore: `setify (setify l) = setify l` for any list `l`.