① A- ~~$\forall a P(\text{Leaf } a)$~~

$\forall a P(\text{Leaf } a) \wedge \forall a_1, a_2.\left[P(a_1) \wedge P(a_2) \rightarrow P(\text{Node } a_1\ a_2)\right]$

$\rightarrow \forall t : \text{Tree}\ P(t)$

B- $\left[\forall i : \text{Int}.\ P(\text{Val } i) \wedge \forall t : \text{Term}.\left[P(t) \rightarrow P(\text{UMinus } t)\right]\right.$

$\left.\wedge\ \forall t_1, t_2 : \text{Term}.\left[P(t_1) \wedge P(t_2) \rightarrow P(\text{Mult } t_1\ t_2)\right]\right]$

$\rightarrow \forall t : \text{Term}.\ P(t)$.

C- i) eval trm = ~~eval~~ -6

rip trm = Mult (Val -3) (Val 2)

pos trm = True

eval (rip trm) = -6

wSign (eval (rip trm)) (pos trm) = -6.

ii) BASE CASE. ~~To show: $\forall i : \text{Int}$~~

To show: $\forall i : \text{Int}\left[\text{eval}(\text{Val } i) = \text{wSign}\ (\text{eval}\ (\text{rip}\ (\text{Val } i)))\ (\text{pos}\ (\text{Val } i))\right]$

RHS = wSign (eval (rip (Val i))) (pos (Val i))

= wSign (eval (rip (Val i))) True — by func def pos.

= eval (rip (Val i)) — by func def wSign.

= eval (Neg i) — by func def rip.

= LHS $\Rightarrow$ BASE CASE PROVEN.

INDUCTIVE STEP.

TO SHOW: $\forall t_1, t_2 : Term. [((eval\ t_1) = wSign\ (eval\ (rip\ t_1))\ (pos\ t_1)) \wedge ((eval\ t_2) = wSign\ (eval\ (rip\ t_2))\ (pos\ t_2)) \longrightarrow (eval\ (Mult\ t_1\ t_2)) = wSign\ (eval\ (rip\ (Mult\ t_1\ t_2)))\ (pos\ (Mult\ t_1\ t_1))]$

INDUCTIVE HYPOTHESIS:

$\forall t_1, t_2 : Term [(eval\ t_1) = wSign\ (eval\ (rip\ t_1))\ (pos\ t_1) \wedge (eval\ t_2) = wSign\ (eval\ (rip\ t_2))\ (pos\ t_2)]$.

$RHS = wSign\ (eval\ (rip\ (Mult\ t_1\ t_2)))\ (pos\ (Mult\ t_1\ t_2))$

$= wSign\ (eval\ (Mult\ (rip\ t_1)\ (rip\ t_2)))\ (pos\ (Mult\ t_1\ t_2))$

By func def rip.

$= wSign\ ((eval\ (rip\ t_1)) * (eval\ (rip\ t_2)))\ (pos\ (Mult\ t_1\ t_2))$

By func def eval

$= wSign\ ((eval\ (rip\ t_1)) * (eval\ (rip\ t_2)))\ ((pos\ t_1) == (pos\ t_2))$

By func def pos

$= (wSign\ (eval\ (rip\ t_1))\ (pos\ t_1)) * wSign\ (eval\ (rip\ t_2))\ (pos\ (rip\ t_2))$

By lemma (C)

$= (eval\ t_1) * (eval\ t_2)$

By inductive hypothesis

$= eval\ (Mult\ t_1\ t_2)$

By func def eval

$= LHS$

$\Rightarrow$ PROVEN BY INDUCTION.

(2) A- Smooth(a) = ~~$\left[\frac{14}{3}, \frac{16}{3}, 5, 4\right]$~~ $[4, 5, 5, 2]$

B- i) $b = []$ (empty int array).

ii) Line 15.

iii) i would be set to 0, it would skip the while loop and then try to access index 0 of the empty array, which does not exist.

iv) //PRE: $a \neq null \wedge a.length > 0$.

C- //INV: $0 \leq i < (a.length - 1) \wedge (i = 0 \rightarrow prv = 0)$

$\wedge (i \neq 0 \rightarrow prv = a[i]) \wedge a[i..a.length) = a_0[i..a.length)$

$\wedge$ ~~$a[0..i)$~~ $\forall k \in [1..i).\ a[k] = \frac{a_0[k-1] + a_0[k] + a_0[k+1]}{3}$

D- GIVEN:

(1) $a \neq null$ — From PRE

(2) $a.length = a_0.length$ — From Assumption.

(3) $0 \leq i < (a.length - 1)$ — From INV

TO SHOW:

(α) $0 \leq i < a.length$

(β) $0 \leq i+1 < a.length$

PROOF:

α follows from (3) and for β:

$0 \leq i < a.length - 1$ by (3)

$1 \leq i+1 < a.length \Rightarrow 0 \leq i+1 < a.length$ adding 1 to inequality.

E-

The first and last elements would tend to half of the element adjacent to them (due to being averaged by zero and the adjacent element). All other elements would tend to a single value (including the first and last elements), therefore all elements would tend to a half of their own element => all elements tend to 0.