1. a)

F calcula la cantidad de elementos iniciales que también aparecen al final de la lista

1. b)

F.[1,2,3,4,1,2]

- El rango según la especificación es:

as = [1,2] as = []

bs = [3,4] ó bs = [1,2,3,4,1,2]

cs = [1,2], cs = []

- La expresión queda: #[] max #[1,2]

#[1,2] max #[]

2 max 0 = 2 ∎

1. c)

$f.xs = \langle \mathrm{Max}\ as,bs,cs : xs = as{+}{+}bs{+}{+}cs \wedge as = cs : \#as \rangle$

$f.xs = \langle \mathrm{Max}\ as,bs : xs = as{+}{+}bs{+}{+}as : \#as \rangle$ } Especificación equivalente por eliminación de cs

• Caso base $[\,] = xs$

$f.[\,] = \langle \mathrm{Max}\ as,bs,cs : [\,] = as{+}{+}bs{+}{+}cs \wedge as = cs : \#as \rangle$

$= \{$Propiedad de listas$\}$

$\langle \mathrm{Max}\ as,bs,cs : [\,] = as = bs = cs : \#as \rangle$

$= \{$Rango unitario + cardinal$\}$

0

$f.[\,] = 0$ ■

• Caso inductivo

$f.(x:xs) = \langle \mathrm{Max}\ as,bs,cs : (x:xs) = as{+}{+}bs{+}{+}cs \wedge as = cs : \#as \rangle$

$= \{$Eliminación de la variable cs$\}$

$\langle \mathrm{Max}\ as,bs : (x:xs) = as{+}{+}bs{+}{+}as : \#as \rangle$ } Especificación equivalente

$= \{$Lógica$\}$

$\langle \mathrm{Max}\ as,bs : (x:xs) = as{+}{+}bs{+}{+}as \wedge (as \neq [\,] \vee as = [\,]) : \#as \rangle$

$= \{$Partición de rango + eliminación de as$\}$ *

$\langle \mathrm{Max}\ bs : (x:xs) = bs : \#[\,] \rangle\ \mathrm{max}\ \langle \mathrm{Max}\ as,bs : x:xs = as{+}{+}bs{+}{+}as \wedge as \neq [\,] : \#as \rangle$

$= \{$Termino constante + cardinal + cambio de variable: $as \to a:as\}$

$0\ \mathrm{max}\ \langle \mathrm{Max}\ a,as,bs : (x:xs) = (a:as){+}{+}bs{+}{+}(a:as) : \#(a:as) \rangle$

$= \{$Propiedades de listas + eliminación de a$\}$

$0\ \mathrm{max}\ \langle \mathrm{Max}\ as,bs : xs = as{+}{+}bs{+}{+}[x]{+}{+}as : \#([x]{+}{+}(as)) \rangle$

$= \{$ Generalizo para poder aplicar H.I $\}$

$g.xs.ks = \langle Max\ as, bs : xs = as ++ bs ++ ks ++ as : \#(ks ++ as) \rangle$

Caso inductivo

$g.(x:xs).ks = \langle Max\ as, bs : (x:xs) = as ++ bs ++ ks ++ as : \#(ks ++ as) \rangle$

$= \{$ Partición de rango + Eliminación de as + Cambio de $as \rightarrow a:as$ $\}$

$\langle Max\ bs : (x:xs) = bs ++ ks : \#ks \rangle\ max\ \langle Max\ a, as, bs : (x:xs) = (a:as) ++ bs ++ ks ++ (a:as) : \#(ks ++ (a:as)) \rangle$

$= \{$ propiedades de listas + eliminación de a $\}$

$\langle Max\ bs : (x:xs) = bs ++ ks : \#ks \rangle\ max\ \langle Max\ as, bs : xs = as ++ bs ++ (ks ++ [x]) ++ as : \#(ks ++ [x] ++ as) \rangle$

$= \{$ Distributiva del $(+)$ sobre el max + Hipótesis $\}$

$\langle Max\ bs : (x:xs) = bs ++ ks : \#ks \rangle\ max\ \ g.xs.(ks ++ [x])$

$= \{$ Tercero excluído y partición de rango $\}$

$\langle Max\ bs : (x:xs) = bs ++ ks \wedge bs = [\,] : \#ks \rangle\ max\ \langle Max\ bs : (x:xs) = bs ++ ks \wedge bs \neq [\,] : \#ks \rangle$

$max\ \ g.xs.(ks ++ [x]) + 1$

$= \{$ Rango unitario y condición + Cambio de variable $\}$

$(ks = xs \rightarrow \#ks,\ ks \neq xs \rightarrow -Inf)\ max\ \langle Max\ b, bs : x = b \wedge xs = bs ++ ks : \#ks \rangle$

$max\ \ g.xs.(ks ++ [x])$

$= \{$ Modularizo $h.xs.ks = \langle Max\ bs : xs = bs ++ ks : \#ks \rangle$, elimino $b = x$ $\}$

$(ks = xs \rightarrow \#ks,\ ks \neq xs \rightarrow -Inf)\ max\ \ h.xs.ks\ \ max\ \ g.xs.(ks ++ [x])$

Scanned with CamScanner

• Caso base

$y.[].ks = \langle \text{Max } as, bs : [] = as ++ bs ++ ks ++ as : \# as \rangle$

$=$ { Propiedades de listas + Rango unitario y condición }

$(ks = [] \to \#[],\ ks \neq [] \to -\text{Inf})$

$y.[].ks = (ks = [] \to 0,$

$ks \neq [] \to -\text{Inf})$ ∎

• Derivo h

Caso base:

$h.[].ks = \langle Max\ bs : [] = bs ++ ks : \#ks \rangle$

$= \{$ Rango unitario y condición $\}$

$(ks = [] \to 0,\ ks \neq [] \to -Inf)$

Caso inductivo

$h.(x:xs).ks = \langle Max\ bs : (x:xs) = bs ++ ks : \#ks \rangle$

$= \{$ Parto el rango en $bs = []\ \vee\ bs \neq []$ $\}$

$= \{$ Rango unitario y condición + cambio de variable $\}$

$(ks = (x:xs) \to \#ks,\ ks \neq (x:xs) \to -Inf)\ max\ \langle Max\ b, bs : (x:xs) = (b:bs) ++ ks : \#ks \rangle$

$= \{$ Propiedades de listas $x = b$, elimino la variable $\}$

$= \{$ Hipótesis $\}$

$(ks = (x:xs) \to \#ks,\ ks \neq (x:xs) \to -Inf)\ max\ h.xs.ks$

Resultado:

$f.xs = g.xs.[]$

$g.[].ks = (\ ks = [] \to 0$
$\quad ks \neq [] \to -Inf)$

$g.(x:xs).ks = (ks = xs \to \#ks\ max\ h.xs.ks\ max\ g.xs.(ks ++ [x]),$
$\quad ks \neq xs \to -Inf\ max\ h.xs.ks\ max\ g.xs.(ks ++ [x]))$

$h.[].ks = (ks = [] \to 0,\ ks \neq [] \to -Inf)$

$h.(x:xs).ks = (\ ks = (x:xs) \to \#ks\ max\ h.xs.ks,$
$\quad ks \neq (x:xs) \to -Inf\ max\ h.xs.ks$

2. a)

$A = [3, -1, 1, -1]$

$n = \langle N i : 0 \leq i \leq M : \langle \Sigma j : i \leq j < M : A.j \rangle < i \rangle$

El rango del N es $i \in \{0, 1, 2, 3, 4\}$

Veo en cada caso si se cumple el término.

- $i = 0$

$$\langle \Sigma j : 0 \leq j < M : A.j \rangle < 0$$
$$\equiv$$
$$3 - 1 + 1 - 1 < 0$$
$$2 < 0$$
$$\equiv$$
$$False$$

- $i = 1$

$$\langle \Sigma j : 1 \leq j < M : A.j \rangle < 1$$
$$\equiv$$
$$-1 < 1$$
$$\equiv$$
$$True$$

- $i = 2$

$$\langle \Sigma j : 2 \leq j < M : A.j \rangle < 2$$
$$\equiv$$
$$0 < 2$$
$$\equiv$$
$$True$$

- $i = 3$

$$\langle \Sigma j : 3 \leq j < M : A.j \rangle < 3$$
$$\equiv$$
$$-1 < 3$$
$$\equiv$$
$$True$$

- $i = 4$

$$\langle \Sigma j : 4 \leq j < M : A.j \rangle < 4$$
$$= \{M = 4, \text{ Rango vacío}\}$$
$$0 < 4$$
$$\equiv$$
$$True$$

Resultado:

$n = 4$

2. b)

Este programa cuenta los segmentos finales de A, incluyendo la lista vacía, en los que la suma de sus elementos es menor al índice o posición del primer elemento del segmento en el array A.

2.C)

Const M : Int;

Var A : array $[0, M)$ of Int; n : Int;

$\{M \geq 0\}$

S

$\{n = \langle N i : 0 \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle\}$

• Propongo empezar refinando un ciclo con previa inicialización:

$\{P\}$

$S0$

$\{I\}$

do $b \rightarrow$

S

od

$\{I\}$

$\{Q\}$

• Propongo el invariante que surge de reemplazar 0 por una nueva variable m en la post-condición:

$\{I : n = \langle N i : m \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle \wedge 0 \leq m \leq M$

• Para la guarda (b) propongo $b = m > 0$

• Probemos que $I \wedge \neg b \Rightarrow Q$

$I \wedge \neg b \equiv n = \langle N i : m \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle \wedge \underbrace{0 \leq m \leq M \wedge 0 \geq m}_{0 = m}$

$= \{$Por leibniz, reemplazando m por $0\}$

$I \wedge \neg b \equiv Q$

$(Q \Rightarrow Q) \equiv$ True

Scanned with CamScanner

- Propongo la cota $t = m$, pues $I \Rightarrow t \geq 0$
- Derivo el cuerpo del ciclo. Pruebo con una asignación múltiple, disminuyendo m en 1, para decrementar la cota.

$\{I \wedge b \wedge t = K\}$

$n, m := E, m-1$

$\{I \wedge t < K\}$

$\equiv$ { Obligación de prueba }

$n = \langle N i : m \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle \wedge 0 < m \leq M \wedge m = K$

$\Rightarrow$

$E = \langle N i : m-1 \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle \wedge 0 \leq m-1 \leq M \wedge m-1 < K$

$\equiv$ { Asumo la hipótesis y pruebo por partes }

- $(0 < m \Rightarrow 0 \leq m-1) \Rightarrow 0 \leq m-1 \leq M$
- $k = m \Rightarrow m-1 < K \equiv m-1 < m \Leftrightarrow \boxed{m < m+1}$
- Despejo E

$\equiv$ { Parto el rango + Rango unitario en $i = m-1$ + Hipótesis (n) }

$E = (\langle \sum j : m-1 \leq j < M : A.j \rangle < i \rightarrow 1, \ (\neg \sum() < i) \rightarrow 0) + n$

$\equiv$ { Parto el rango en $j = m-1$ }

$E = (\langle \sum j : m \leq j < M : A.j \rangle + A.(m-1) < m-1 \rightarrow 1, \ \neg(\sum() + A.(m-1) < m-1) \rightarrow 0) + n$

- Propongo fortalecer nuestro invariante introduciendo una nueva variable: $s = \langle \sum j : m \leq j < M : A.j \rangle$.
- Dado el análisis por casos al que llegamos, propongo introducir un condicional con estos casos como guardas, asignando los valores que corresponden por caso:

$\{I \wedge b \wedge t = k\}$

if $s + A.(m-1) < m-1 \rightarrow$

  $S1$

[] $s + A.(m-1) \geq m-1 \rightarrow$

  $S2$

fi

• $\forall x, y \in \mathbb{R}, x \geq y \lor x < y$

$\Rightarrow$ siempre se cumple alguna guarda

• Probemos ambas ternas de la obligación de prueba y despejemos las asignaciones para n y s:

$\{I \land b \land s + A.(m-1) < m-1 \land t = k\}$

$n, s, m := E, F, m-1$ $\boxed{\text{Primera terna}}$

$\{I \land b < t = k\}$

$= \{\text{Repitiendo pasos anteriores}\}$

$E = (s + A.(m-1) < m-1 \to 1,\ s + A.(m-1) \geq m-1 \to 0) + n \land$

$F = \langle \sum j : m-1 \leq j < M : A.j \rangle$

$= \{\text{Resuelvo análisis por caso y parto el rango}\}$

$E = 1 + n \land F = \langle \sum j : j = m-1 : A.j \rangle + \langle \sum j : m \leq j < M : A.j \rangle$

$= \{\text{Rango unitario + hipótesis}(s)\}$

$\boxed{E = 1 + n \ \land \ F = A.(m-1) + s}$

$\{I \land b \land s + A.(m-1) \geq m-1 \land t = k\}$

$n, s, m := E, F, m-1$ $\boxed{\text{Segunda terna}}$

$\{I \land t < k\}$

$= \{\text{Mismos pasos que la otra terna}\}$

$E = n \ \land \ F = A.(m-1) + s$

Entonces, el cuerpo del ciclo es:

$\{I \wedge b \wedge t = k\}$

if $s + A.(n-1) < m-1 \rightarrow$

$n, s, m := 1+n,\ s + A.(n-1),\ m-1$

$[]\ s + A.(n-1) \geq m-1 \rightarrow$

$n, s, m := n,\ s + A.(n-1),\ m-1$

fi

$\{I \wedge t < k\}$

- Derivo la inicialización, pruebo asignándole M a m

$\{M \geq 0\}$

$n, s, m := E, F, M$

$\{I\}$

$= \{$Obligación de prueba$\}$

$M \geq 0$

$\Rightarrow$

$E = \langle N i : M \leq i \leq M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle \wedge F = \langle \sum j : M \leq j < M : A.j \rangle$

$= \{M \leq i \leq M \Rightarrow i = M$; Rango unitario y rango vacío$\}$

$E = (0 < M \rightarrow 1,\ 0 \geq M \rightarrow 0) \wedge F = 0$

- Planteo cambiar nuestra inicialización por un condicional que contemple el análisis por casos, asignando lo que corresponda
- Igual que antes, siempre se cumplirá una de las guardas.
- Probemos ambas ternas de la obligación de prueba.

$\{M \geq 0 \wedge M > 0\}$

$n, s, m := E, S, M$

$\{I\}$

$= \{$Uso los pasos anteriores + resuelvo el análisis por casos$\}$

$\boxed{E = 1,\ F = 0}$

Primera terna

$\{M \geq 0 \wedge M \leq 0\}$

$n, s, m := E, S, M$

$\{I\}$

$= \{$Igual que con la otra terna$\}$

$\boxed{E = 0,\ F = 0}$

Segunda terna

Nuestro programa queda:

```
Const M: Int;
Var A: array [0,M) of Int, n: Int, s: Int, m: Int
{M >= 0}
 if M <= 0 ->
   n, s, m := 0, 0, M
 [] M > 0 ->
   n, s, m := 1, 0, M
 fi
do m > 0 ->
 if s + A.(m-1) < m-1
   n, s, m := 1+n, s + A.(m-1), m-1
 [] s + A.(m-1) >= m-1
   n, s, m := n, s+A.(m-1), m-1
 fi
od
```

$\{n = \langle N i : 0 \leq i < M : \langle \sum j : i \leq j < M : A.j \rangle < i \rangle$

3.

Const N: Int, K: Int;

Var r: Bool, A: array $[0, N)$ of Int;

$\{N \geq 2\}$

S

$\{r = \langle \exists i : 0 < i < N : A.(i-1) - A.i < K \rangle\}$

Por la presente declaro que la resolución de este examen es obra de mi exclusiva autoría y respetando las pautas y criterios fijados en los enunciados. Asimismo, declaro conocer el régimen de infracción de los estudiantes cuyo texto ordenado se encuentra en el apéndice de la Res. Rec. 1554/2018.