DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico N°1

Especificación y WP

18 de septiembre de 2023 — Algoritmos y Estructuras de Datos

RexonaNoTeAbandona

Integrante	LU	Correo electrónico
Lamonica, Ivo	66/22	ivolamonicam@gmail.com
Bravo, Santiago	750/22	santolau2013@gmail.com
Masetto, Lautaro	1052/22	lemasetto@gmail.com
Corales, Manuel	382/22	maugcorales@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Problemas

1. **hayBallotage**

   proc hayBallotage (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : Bool

   requiere $\{|escrutinio| \geq 2\}$

   requiere $\{valoresNoNegativos(escrutinio)\}$

   requiere $\{noHayEmpate(escrutinio)\}$

   requiere $\{existePartidoConVotos(escrutinio)\}$

   asegura $\{res = false \leftrightarrow porcentajePrimero(escrutinio) > 0,45\}$

   asegura $\{res = true \leftrightarrow porcentajePrimero(escrutinio) > 0,4 \wedge$
   $((porcentajePrimero(escrutinio) - porcentajeSegundo(escrutinio)) < 0,1) \vee$
   $porcentajePrimero(escrutinio) < 0,4\}$

2. **hayFraude**

   proc hayFraude (in escrutinio_presidencial : $seq\langle\mathbb{Z}\rangle$, in escrutinio_senadores : $seq\langle\mathbb{Z}\rangle$, in escrutinio_diputados : $seq\langle\mathbb{Z}\rangle$) : Bool

   requiere $\{$
   $|escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados|\}$

   requiere $\{$
   $valoresNoNegativos(escrutinio\_presidencial) \wedge$
   $valoresNoNegativos(escrutinio\_senadores) \wedge$
   $valoresNoNegativos(escrutinio\_diputados)$
   $\}$

   requiere $\{$
   $existePartidoConVotos(escrutinio\_presidencial) \wedge$
   $existePartidoConVotos(escrutinio\_senadores) \wedge$
   $existePartidoConVotos(escrutinio\_diputados)$
   $\}$

   asegura $\{res = false \leftrightarrow$
   $sumaDeVotos(escrutinio\_senadores) = sumaDeVotos(escrutinio\_presidencial) \wedge$
   $sumaDeVotos(escrutinio\_presidencial) = sumaDeVotos(escrutinio\_diputados)$
   $\}$

   Comentarios: Aclararon que podemos asumir que las 3 listas siempre tienen el mismo largo

3. **obtenerSenadoresEnProvincia**

   proc obtenerSenadoresEnProvincia (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ x $\mathbb{Z}$

   requiere $\{valoresNoNegativos(escrutinio)\}$

   requiere $\{|escrutinio| > 2\}$

   requiere $\{noHayEmpate(escrutinio)\}$

   requiere $\{existePartidoConVotos(escrutinio)\}$

   asegura $\{(\exists i, j : \mathbb{Z})\ (0 \leq i, j < |escrutinio| - 1) \wedge$
   $i \neq j \longrightarrow_L (res_0 = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i] \wedge$
   $res_1 = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j])\}$

4. **calcularDHondtEnProvincia**

   proc calcularDHondtEnProvincia (in cant_bancas : $\mathbb{Z}$, in escrutinio :$seq\langle\mathbb{Z}\rangle$) : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$

   requiere $\{cant\_bancas > 0\}$

   requiere $\{valoresNoNegativos(escrutinio)\}$

requiere $\{|escrutinio| > 1\}$

requiere $\{noHayEmpate(escrutinio)\}$

requiere $\{existePartidoConVotos(escrutinio)\}$

asegura $\{|res| = |escrutinio| - 1\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L (|res[i]| = cant\_bancas)\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| \longrightarrow_L (porcentajeVotos(escrutinio[i], escrutinio)) < 0,03 \leftrightarrow |res[i]| = 0\}$

asegura $\{(\forall i, j, k, l : \mathbb{Z})((0 \leq i, k < |res| \wedge_L 0 \leq j, l < cant\_bancas \wedge_L i \neq k \wedge_L |res[i]| \neq 0 \wedge_L |res[k]| \neq 0) \longrightarrow_L res[i][j] \neq res[k][l]\}$

asegura $\{(\forall i, j : \mathbb{Z})(0 \leq i < |res| \wedge_L 0 \leq j < cant\_bancas \wedge_L res[i][j] = divisionEntera(escrutinio[i], j + 1)\}$

5. **obtenerDiputadosEnProvincia**

proc obtenerDiputadosEnProvincia (in cant_bancas : $\mathbb{Z}$, in escrutinio :$seq\langle\mathbb{Z}\rangle$, in dHondt: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $seq\langle\mathbb{Z}\rangle$

requiere $\{cant\_bancas > 0\}$

requiere $\{|dHondt| > 0\}$

requiere $\{valoresNoNegativos(escrutinio)\}$

requiere $\{|escrutinio| > 1\}$

requiere $\{noHayEmpate(escrutinio)\}$

requiere $\{existePartidoConVotos(escrutinio)\}$

requiere $\{(\forall i : \mathbb{Z})(0 \leq i < |dHondt| \longrightarrow_L (|dHondt[i]| = cant\_bancas \vee |dHondt[i]| = 0))\}$

asegura $\{valoresNoNegativos(res)\}$

asegura $\{\sum_{i=0}^{|res|-1} res[i] = cant\_bancas\}$

Comentarios:

En el séptimo requiere consideramos que el largo de las listas de los partidos es igual a cantidad de bancas ya que el caso límite es que dHondt tenga una única lista, y todos los cocientes sean las bancas a cubrir; tambien puede ser cero pues, en el ejercicio anterior especificamos que en el caso de que un partido no alcanzara el umbral, al pasarlo a la matriz dHondt se pase como una lista vacia.

6. **validarListasDiputadosEnProvincia**

proc validarListasDiputadosEnProvincia (in cant_bancas : $\mathbb{Z}$, in listas :$seq\langle seq\langle dni : \mathbb{Z}\,\mathrm{x}\, genero : \mathbb{Z}\rangle\rangle$) : Bool

requiere $\{cant\_bancas > 0\}$

requiere $\{|listas| > 0\}$

requiere $\{dni > 0\}$

requiere $\{genero = 1 \vee genero = 2\}$

requiere $\{(\forall i : \mathbb{Z})(0 \leq i < |listas| \wedge_L 0 < |listas[i]|)\}$

asegura $\{res = true \leftrightarrow (\forall i : \mathbb{Z})(0 \leq i < |listas| \longrightarrow_L cant\_bancas = |listas[i]|)\}$

asegura $\{alternanciaDeGenero(cant\_bancas, listas)\}$

1.1. Funciones y predicados auxiliares

`aux sumaDeVotos` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|escrutinio|-1} escrutinio[i]$;

`aux votosPrimero` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|escrutinio|-2}$if partidoMasVotado(escrutinio, escrutinio[i]) then escrutinio[i] else 0 fi ;

`aux votosSegundo` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|escrutinio|-2}$if segundoPartidoMasVotado(escrutinio, escrutinio[i]) then escrutinio[i] else 0 fi ;

`aux porcentajeVotos` (votosPartido: $\mathbb{Z}$, escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= votosPartido/sumaDeVotos(escrutinio)$;

`aux porcentajePrimero` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= votosPrimero(escrutinio)/sumaDeVotos(escrutinio)$;

`aux porcentajeSegundo` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= votosSegundo(escrutinio)/sumaDeVotos(escrutinio)$;

`aux divisionEntera` (dividendo: $\mathbb{Z}$, divisor: $\mathbb{Z}$) : $\mathbb{Z}$ $= (dividendo -$ (dividendo mod divisor)$)/divisor$;

`pred valoresNoNegativos` (escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i:\mathbb{Z})(0 < i < |escrutinio| \longrightarrow_L 0 \leq escrutinio[i])$
}

`pred existePartidoConVotos` (escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\exists i:\mathbb{Z})\ (0 \leq i < |escrutinio| - 1 \longrightarrow_L escrutinio[i] > 0)$
}

`pred partidoMasVotado` (escrutinio: $seq\langle\mathbb{Z}\rangle$, primero: $\mathbb{Z}$) {
$(\forall i:\mathbb{Z})(0 \leq i < |escrutinio| - 1 \longrightarrow_L escrutinio[i] \leq primero)$
}

`pred segundoPartidoMasVotado` (escrutinio: $seq\langle\mathbb{Z}\rangle$, segundo: $\mathbb{Z}$) {
$(\exists k:\mathbb{Z})\ ((\forall i:\mathbb{Z})((0 \leq i, k < |escrutinio| - 1 \wedge_L i \neq k) \longrightarrow_L escrutinio[i] < segundo < escrutinio[k]))$
}

`pred noHayEmpate` (escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i:\mathbb{Z})((0 \leq i, j < |escrutinio| \wedge_L i \neq j) \longrightarrow_L escrutinio[i] \neq escrutinio[j])$
}

`pred alternanciaDeGenero` (in cant_bancas : $\mathbb{Z}$, in listas :$seq\langle seq\langle dni : \mathbb{Z}\,\mathrm{x}\, genero : \mathbb{Z}\rangle\rangle$) {
$(\forall i, j:\mathbb{Z})((0 \leq i < |listas| \wedge 0 \leq j < |cant\_bancas| - 1 \longrightarrow_L$
$if\ (listas[i][j]_1 == 1)\ then\ (listas[i][j+1]_1 == 2)\ else\ (listas[i][j+1]_1 == 1)$
}

2. Implementaciones y demostraciones de correctitud

2.1. Algoritmos

hayBallotage (in escrutinio: $\mathbb{Z}$): Bool

```
res := false;
i := 0;
sumaVotos := 0
votosPrimero := 0;
votosSegundo := 0;

while (i < escrutinio.size() − 1) do
    sumaVotos := sumaVotos + escrutinio[i];
    if (escrutinio[i] > votosPrimero) then
        votosSegundo := votosPrimero;
        votosPrimero := escrutinio[i];
    else
        if(escrutinio[i] > votosSegundo) then
            votosSegundo := escrutinio[i];
        else
            skip
    i := i + 1;
endwhile

sumaVotos := sumaVotos + escrutinio[i];
porcentajePrimero := votosPrimero/sumaVotos;
porcentajeSegundo := votosSegundo/sumaVotos;

if (porcentajePrimero < 0.45 &&
    porcentajePrimero > 0.4 && ((porcentajePrimero − porcentajeSegundo) < 0.1)) ||
    porcentajePrimero < 0.4  then
    res := true;
else
    skip
endif
```

hayFraude(in escrutinio_senadores : $seq\langle\mathbb{Z}\rangle$, in escrutinio_diputados : $seq\langle\mathbb{Z}\rangle$, in escrutinio_presidencial : $seq\langle\mathbb{Z}\rangle$): Bool

```
res := false;
sumaVotoSenadores := 0;
sumaVotosDiputados := 0;
sumaVotosPresidente := 0;
i := 0;
while (i < escrutinio_senadores.size()) do
        sumaVotoSenadores := sumaVotoSenadores + escrutinio_senadores[i];
        sumaVotoDiputados := sumaVotoDiputados + escrutinio_diputados[i];
        sumaVotosPresidente := sumaVotoPresidente + escrutinio_presidencial[i];
        i := i + 1;
endwhile
if (sumaVotosSenadores == sumaVotosDiputados == sumaVotosPresidente)  then
    res := true;
else
    skip
endif
```

obtenerSenadoresEnProvincia (in escrutinio: $seq\langle\mathbb{Z}\rangle$): $\mathbb{Z} \times \mathbb{Z}$

```
fst := 0;
snd := 0;
res := [0,0];
i := 0;
if (escrutinio[0]<escrutinio[1]) then
    fst:= 1;
else
    snd:= 1;
endif
while (i < escrutinio.size()−1) do
        if (escrutinio[i] > escrutinio[fst]) then
            snd := fst;
            fst := i;
        else
            if (escrutinio[i] > escrutinio[snd]) then
                snd := i;
            else
                skip
        i := i + 1;
endwhile
res[0] := fst;
res[1] := snd;
```

Comentarios: Como en SmallLang no existe la declaración o asignación de tuplas, tratamos al res como un array, posteriormente haciendo las asignaciones pertinentes. También teniendo en cuenta que luego para verificar la correctitud la definición de esta inicialización la vamos a hacer diciendo que el largo de res tiene que ser mayor a 1.

validarListasDiputadosEnProvincia (in escrutinio: $\mathbb{Z}$): Bool

```
res := true;
i := 0;

while (i < |listas|) do
    if (cant_bancas != listas[i].size()) then
        res := false;
    else
        j := 0;
        while (j < cant_bancas − 1) do
            if (listas[i][j][1] == listas[i][j + 1][1]) then
                res := false;
            else
                skip
            j := j + 1;
           endwhile

i := i + 1;
endwhile
```

2.2. Demostraciones de correctitud

2.2.1. hayFraude

Sea **S** el programa completo, dividimos el código en 3 partes:

S1:

```
res := True;
sumaVotoSenadores := 0;
sumaVotosDiputados := 0;
sumaVotosPresidente := 0;
```

C:

```
while (i < escrutinio_senadores.size()) do
        sumaVotoSenadores := sumaVotoSenadores + escrutinio_senadores[i];
        sumaVotoDiputados := sumaVotoDiputados + escrutinio_diputados[i];
        sumaVotosPresidente := sumaVotosPresidente + escrutinio_presidencial[i];
        i := i + 1;
endwhile
```

S2:

```
if (sumaVotosSenadores == sumaVotosDiputados == sumaVotosPresidente)   then
    res:= False;
else
    skip
endif
```

Idea para probar la correctitud: queremos llegar a que $Pre \implies wp(S, Post)$ mediante el corolario de monotonía

Sean Pc y Qc la precondición y postcondicion de C respectivamente, si tenemos que:

- $Pre \implies wp(S1, Pc)$
- $Pc \implies wp(C, Qc) \equiv \{Pc\}\ C\ \{Qc\}$
- $Qc \implies wp(S2, Post)$

Entonces podremos probar que vale $\{Pre\}\ S\ \{Post\}$, es decir, el programa es correcto.

Primeramente planteamos quienes son Pc y Qc

$Pre \equiv$
$(\forall i : \mathbb{Z})(0 \leq i < |escrutinio\_senadores| \longrightarrow_L escrutinio\_senadores[i] \geq 0)$

$Pc \equiv$
$Pre \wedge_L (sumaVotosSenadores = 0 \wedge sumaVotosDiputados = 0 \wedge sumaVotosPresidente = 0 \wedge$
$i = 0 \wedge res = False)$

$Qc \equiv$
$sumaVotosSenadores = \sum_{j=0}^{|escrutinio\_senadores|-1} escrutinio\_senadores[j] \quad \wedge$
$sumaVotosDiputados = \sum_{j=0}^{|escrutinio\_senadores|-1} escrutinio\_diputados[j] \quad \wedge$
$sumaVotosPresidente = \sum_{j=0}^{|escrutinio\_diputados|-1} escrutinio\_presidencial[j] \quad \wedge$
$i = |escrutinio\_senadores|$

Δ Primera parte:

► Comenzamos probando que $Pre \implies wp(S1, Pc)$

Pre $\implies wp(S1, Pc) \equiv \quad i = 0 \quad \wedge \quad sumaVotosSenadores = 0 \quad \wedge \quad sumaVotosDiputados = 0 \quad \wedge$ $sumaVotosPresidente = 0 \ \wedge \ res = False$
$\equiv \ 0 = 0 \ \wedge \ 0 = 0 \ \wedge \ 0 = 0 \ \wedge \ 0 = 0 \ \wedge \ False = False$
$\equiv \ True \ \wedge \ True \ \wedge \ \ True \ \wedge \ \ True \ \wedge \qquad True \qquad \equiv \qquad True$

Δ Segunda parte:

► Ahora vamos a demostrar que $Pc \implies wp(C, Qc) \equiv \{Pc\}\ C\ \{Qc\}$. Para ello vamos a utilizar el teorema de correctitud de un ciclo, que dice que una tripla de Hoare es válida si:

$$\left.\begin{array}{l} 1.\ Pc \Rightarrow I \\ 2.\ \{I \wedge B\}\, C\, \{I\} \\ 3.\ I \wedge \neg B \Rightarrow Qc \end{array}\right\} \text{Teorema del Invariante} \quad \left.\begin{array}{l} 4.\ \{I \wedge B \wedge v_0 = fv\}\, C\, \{fv < v_0\} \\ 5.\ I \wedge fv \leq 0 \Rightarrow \neg B \end{array}\right\} \text{Teorema de Terminación}$$

► A continuación, vamos a definir el invariante, la funcion variante y demostrar cada una de las condiciones para que la tripla de Hoare sea válida.

$I \equiv$
$0 \leq i \leq |escrutinio\_senadores| \ \wedge \ sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \ \wedge \ sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \ \wedge \ sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]$

fv = $|escrutinio\_senadores|$ - i

• $PC \Rightarrow I$

Para demostrar esto, reemplazo en el invariante los valores que estan en PC:

$I \equiv$
$0 \leq 0 \leq |escrutinio\_senadores| \ \wedge \ 0 = \sum_{j=0}^{0-1} escrutinio\_senadores[j] \ \wedge \ 0 = \sum_{j=0}^{0-1} escrutinio\_diputados[j] \ \wedge$
$0 = \sum_{j=0}^{0-1} escrutinio\_presidencial[j]$

$\equiv 0 \leq |escrutinio\_senadores| \ \wedge \ 0 = \sum_{j=0}^{-1} escrutinio\_senadores[j] \ \wedge \ 0 = \sum_{j=0}^{-1} escrutinio\_diputados[j] \ \wedge$
$0 = \sum_{j=0}^{-1} escrutinio\_presidencial[j]$

$\equiv 0 \leq |escrutinio\_senadores| \ \wedge \ 0 = 0 \ \wedge \ 0 = 0 \ \wedge \ 0 = 0$

$\equiv True \ \wedge \ True \ \wedge \ True \ \wedge \ True$

$\equiv True$

► Asi queda demostrado que se cumple la implicación ✓

• $\{I \wedge B\}\, C\, \{I\} \equiv (I \wedge B) \implies wp(C, I)$

► Entonces, calculo la wp(C, I):

wp(sumaVotosSenadores = sumaVotosSenadores + escrutinio _senadores[i]; sumaVotosDiputados = sumaVotosDiputados + escrutinio _diputados[i]; sumaVotosPresidente = sumaVotosPresidente + escrutinio _presidente[i]; i = i + 1, I)

$\equiv wp(sumaVotosSenadores = sumaVotosSenadores + escrutinio\_senadores[i],\ wp(sumaVotosDiputados = sumaVotosDiputados + escrutinio\_diputados[i],\ wp(sumaVotosPresidente = sumaVotosPresidente + escrutinio\_presidente[i],\ wp(i := i + 1, I))))$

► Vamos a ir por partes

wp(i := i + 1, I) $\equiv$ $True \wedge_L (0 \leq i+1 \leq |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i+1-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i+1-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i+1-1} escrutinio\_presidencial[j])$

$\equiv -1 \leq i \leq |escrutinio\_senadores| - 1 \wedge sumaVotosSenadores = \sum_{j=0}^{i} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i} escrutinio\_presidencial[j]$

wp(sumaVotosPresidente := sumaVotosPresidente + escrutinio _presidencial[i], wp(i := i + 1, I))

$\equiv \quad 0 \leq i < |escrutinio\_presidencial| \wedge_L (-1 \leq i < |escrutinio\_senadores| \wedge$

$sumaVotosSenadores = \sum_{j=0}^{i} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i} escrutinio\_diputados[j] \wedge$

$sumaVotosPresidente + escrutinio\_presidencial[i] = \sum_{j=0}^{i} escrutinio\_presidencial[j])$

► Como la longitud de las listas de escrutinios son iguales, hago la intersección entre los dos rangos de i, quedando: $0 \leq i < |escrutinio\_senadores|$

$\equiv \quad 0 \leq i < |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i} escrutinio\_diputados[j] \wedge sumaVotosPresidente + escrutinio\_presidencial[i] = \sum_{j=0}^{i} escrutinio\_presidencial[j]$

$\equiv \quad 0 \leq i < |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i} escrutinio\_presidencial[j] - escrutinio\_presidencial[i]$

$\equiv \quad 0 \leq i < |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]$

► Repito el mismo proceso con los S (asignaciones) que faltan y nos queda que:

wp(C, I) $\equiv \quad 0 \leq i < |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]$

► Luego de simplificar, nos falta probar que

(I $\wedge B) \implies wp(C, I)$

$0 \leq i \leq |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge i < |escrutinio\_senadores|$

$\implies wp(C, I)$

$0 \leq i < |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \implies wp(C, I)$

► Esto es trivialmente cierto, por lo que podemos concluir que $\{I \wedge B\}\, C\, \{I\}$ es una tripla de Hoare válida ✓

- $(I \wedge \neg B) \implies Qc$

I $\wedge \neg B \equiv 0 \leq i \leq |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge i \geq |escrutinio\_senadores|$

$\equiv i = |escrutinio\_senadores| \wedge sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]$

$\equiv sumaVotosSenadores = \sum_{j=0}^{|escrutinio\_senadores|-1} escrutinio\_senadores[j] \quad \wedge$
$sumaVotosDiputados = \sum_{j=0}^{|escrutinio\_senadores|-1} escrutinio\_diputados[j] \quad \wedge$
$sumaVotosPresidente = \sum_{j=0}^{|escrutinio\_senadores|-1} escrutinio\_presidencial[j] \quad \wedge$
$i = |escrutinio\_senadores|$

▶ Por lo tanto, tenemos que $(I \wedge \neg B) \implies Qc$ ✓

• $\{I \wedge B \wedge v_0 = fv\}\, C\, \{fv < v_0\}$

Para probar $\{I \wedge \neg B \wedge v_0 = |s| - i\}C\{fv < v_0\}$, tenemos que probar que
$\{I \wedge \neg B \wedge v_0 = |escrutinio\_senadores| - i\} \implies wp(C, |escrutinio\_senadores| - i < v_0)$

$wp(C, |escrutinio\_senadores| - i < v_0) \equiv wp(Sa; i := i+1, |escrutinio| - i < v_0) \equiv wp(Sa, wp(i := i+1,$
$wp(C, |escrutinio\_senadores| - i < v_0)))$
(En este caso llamo Sa a las asignaciones de suma de votos de presidente, senadores y diputados)

$wp(i := i+1, |escrutinio\_senadores| - i < v_0) \equiv True \wedge_L |escrutinio\_senadores| - (i+1) < v_0$
$wp(i := i+1, |escrutinio\_senadores| - i < v_0) \equiv True \wedge_L |escrutinio\_senadores| - i - 1) < v_0$

Ahora dado que para en todo Sa no se realiza ninguna asignación de la variable i, al aplicar los axiomas no tendremos que hacer ningún reemplazo esta variable, por lo tanto, la parte que queremos saber de la wp ya la tenemos.

$wp(C, |escrutinio\_senadores| - i < v_0) \equiv |escrutinio\_senadores| - i - 1 < v_0$

De I $\wedge \neg B \wedge v_0 = fv$ busco la parte que necesito para probar esta implicación:

I $\wedge \neg B \wedge v_0 = fv \equiv \quad ... \quad v_0 = |escrutinio\_senadores| - i$

Reemplazo en la wp a v_0:
$|escrutinio\_senadores| - i - 1 < v_0 \quad \equiv \quad |escrutinio\_senadores| - i - 1 < |escrutinio\_senadores| - i \equiv True$

▶ Por lo tanto: $\{I \wedge \neg B \wedge v_0 = fv\}C\{fv < v_0\}$ ✓

• I $\wedge fv \leq 0 \implies \neg B$

I $\wedge fv \leq 0 \equiv 0 \leq i \leq |escrutinio\_senadores| \quad \wedge \quad sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \quad \wedge$
$sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \quad \wedge \quad sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \quad \wedge$
$|escrutinio\_senadores| - i \leq 0$

I $\wedge fv \leq 0 \equiv 0 \leq i \leq |escrutinio\_senadores| \quad \wedge \quad sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \quad \wedge$
$sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \quad \wedge \quad sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \quad \wedge$
$|escrutinio\_senadores| \leq i$

I $\wedge fv \leq 0 \equiv i = |escrutinio\_senadores| \quad \wedge \quad sumaVotosSenadores = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \quad \wedge$
$sumaVotosDiputados = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \quad \wedge \quad sumaVotosPresidente = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]$

$\neg B \equiv i \geq |escrutinio\_senadores|$

Esta implicacion es True ya que si $\quad i = $ |escrutinio_senadores| $\quad$ también se cumple que
|escrutinio_senadores| $\geq$ |escrutinio_senadores|

▶ Por lo tanto: $I \wedge fv \leq 0 \implies \neg B \quad$ ✓

Δ Tercera parte:

Nos queda por ver:

- $QC \Rightarrow wp(S2, Post)$

Recuerdo: S2 = if (sumaVotosDiputados == sumaVotosSenaores == smaVotosPresidente) then res := False else skip
Para encontrar la wp necesitamos que se cumpla def(S2).
Para esto llamo B a (sumaVotosDiputados == sumaVotosSenaores == smaVotosPresidente) y llamo R1 a res := False.
Notar que B es igual al segundo término del Post.

$wp(S2, Post) \equiv def(B) \longrightarrow_L (((B \wedge wp(R1, Post)) \vee (\neg B \wedge wp(skip, Post))))$

Ahora bien, en el consecuente tengo dos términos separados con un $\vee$, trabajemos con el primer término.

$(B \wedge wp(R1, post))$

Este término nos dice que se esta cumpliendo B y necesitamos buscar la wp(R1,Post).
$wp(R1, Post) \equiv def(R1) \wedge Post^{res}_{True} \equiv Post^{res}_{True}$
Entonces. Tenemos que se esta cumpliendo B y ademas, res cambia a False entonces tenemos $False = False \iff True$.
Esto nos da como resultado True. Esto concluye el primer término.

Vamos con el segundo:

$(\neg B \wedge wp(skip, Post))$ luego $wp(skip, Post) \equiv Post$

Cuando $\neg B$ ver que $QC \implies wp(S2, Post) \equiv QC \implies Post$

Entonces de este termino podemos ver que se esta cumpliendo $\neg B$, es decir, que la longitud de las sumas es distinta, y ademas, res = True. Quedando en el Post: $True = False \iff False$, esto nos devueleve True que es lo que buscamos.
Con estos pasos demostramos que nuestro if esta correcto y que tambien se cumple $QC \longrightarrow wp(S2, Post)$.

Finalmente queda demostrado que el programa es correcto debido al corolario de monotonia, ya que pudimos mostrar que:
{Pre} S {Post}, donde S contiene a S1, Pc, C, Qc, S2.
Entonces $\{Pre\}\ S\ \{Post\} Pre \longrightarrow wp(S1, Pc, C, Qc, S2, Post)$

2.2.2. obtenerSenadoresEnProvincia

Sea **S** el programa completo, dividimos el código en 3 partes:

S1:

```
fst := 0;
snd := 0;
res := [0,0];
i := 0;
if (escrutinio[0] < escrutinio[1]) then
    fst := 1;
else
    snd := 1;
endif
```

C:

```
while (i < escrutinio.size()-1) do
        if (escrutinio[i] > escrutinio[fst]) then
            snd := fst;
            fst := i;
        else
            if (escrutinio[i] > escrutinio[snd]) then
                snd := i;
            else
                skip
        i := i + 1;
endwhile
```

S2:

```
res[0] := fst;
res[1] := snd;
```

Tenemos que $S \equiv S1; C; S2$ y la precondición y postcondición(de ahora en más *pre* y *post*)
Idea para probar la correctitud: queremos llegar a que $pre \implies wp(S, post)$ mediante el corolario de monotonía

Sean Pc y Qc la precondición y postcondicion de C respectivamente, si tenemos que:

- $Pre \implies wp(S1, Pc)$
- $Pc \implies wp(C, Qc) \equiv \{Pc\}C\{Qc\}$
- $Qc \implies wp(S2, Post)$

Entonces podremos probar que vale $\{Pre\}S\{Post\}$, es decir, el programa es correcto.

Empecemos con el caso de $Qc \implies wp(S2, Post)$:

$wp(S2, Post) \equiv$ wp(setAt(res,0,fst);setAt(res,1,snd), Post)$\equiv$ wp(setAt(res,0,fst), wp(setAt(res,1,snd), Post))

Primera parte:

$wp(setAt(res, 1, snd), Post) \equiv \quad 1 < |res| \wedge_L (\exists i, j : \mathbb{Z})\ (0 \leq i, j < |escrutinio| - 1) \wedge$
$i \neq j \longrightarrow_L setAt(res, 1, snd)_0 = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i] \wedge$
$setAt(res, 1, snd)_1 = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j])$

$wp(setAt(res,1,snd),Post) \equiv \quad 1 < |res| \wedge_L (\exists i,j:\mathbb{Z})\ (0 \leq i,j < |escrutinio|-1) \wedge$
$i \neq j \longrightarrow_L res_0 = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i] \wedge$
$snd = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j])$

Comentarios: Cuando hacemos el setAt, no separamos los casos donde el índice es igual al segundo elemento del setAt (es decir setAt(res,ESTE,snd)), porque el índice siempre va a ser exactamente el número que está en la Post, es decir, no podría pasar que 1 sea distinto de 1, entonces no separamos esos casos.

Segunda Parte:

$wp(setAt(res,0,fst), wp(setAt(res,1,snd),Post)) \equiv \ 0 < |res| \ \wedge_L \ 1 < |res| \wedge_L (\exists i,j:\mathbb{Z})\ (0 \leq i,j < |escrutinio|-1) \wedge$
$i \neq j \longrightarrow_L setAt(res,0,fst)_0 = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i] \wedge$
$snd = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j])$

$wp(setAt(res,0,fst), wp(setAt(res,1,snd),Post)) \equiv \quad 1 < |res| \wedge_L (\exists i,j:\mathbb{Z})\ (0 \leq i,j < |escrutinio|-1) \wedge$
$i \neq j \longrightarrow_L fst = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i] \wedge$
$snd = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j])$

Esto último es equivalente a $wp(S2, Post)$
Por lo tanto, para que esta implicación sea verdadera, requerimos que Qc sea exactamente eso

Definimos a Qc:

$Qc \quad \equiv \quad 1 < |res| \quad \wedge_L \quad ((\exists i,j:\mathbb{Z})\ (0 \leq i,j < |escrutinio|-1) \quad \wedge \quad i \neq j \quad \longrightarrow_L$
$((fst = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i]) \quad \wedge \quad (snd = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j]))$

Por lo tanto siendo Qc igual a la $wp(S2, Post)$, podemos afirmar que $Qc \implies wp(S2, Post)$ ✓ es True.

Para probar que se cumple la siguiente tripla de Hoare $\{Pc\}C\{Qc\}$, es decir, $Pc \implies wp(C, Qc)$, usamos el Teorema del Invariante, y el Teorema de Terminación para probar que el ciclo termina.

Teorema del invariante:

- $Pc \implies I$
- $\{I \wedge B\}S\{I\}$
- $I \wedge \neg B \implies Qc$

• $Pc \implies I$
Definimos el Invariante y la Pc (en base a la Pre):

I $\equiv \quad 0 \leq i \leq |escrutinio|-1 \quad \wedge_L \quad 0 \leq fst, snd \leq i \quad \wedge_L \quad escrutinio[fst] > escrutinio[snd] \quad \wedge_L \quad 1 < |res| \quad \wedge_L$ $((\forall j:\mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \quad \wedge \quad (\forall k:\mathbb{Z})((0 \leq k < i \wedge k \neq fst) \longrightarrow_L escrutinio[snd] \geq escrutinio[k])) \wedge (\forall p,m:\mathbb{Z})((0 \leq p,m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

$Pre \equiv \quad (\forall i:\mathbb{Z})(0 < i < |escrutinio| \longrightarrow_L 0 \leq escrutinio[i]) \quad \wedge \quad |escrutinio| > 2 \quad \wedge$

$(\forall p,j:\mathbb{Z})((0\leq p,j<|escrutinio|\wedge_L p\neq j)\longrightarrow_L escrutinio[p]\neq escrutinio[j])\wedge$
$(\exists k:\mathbb{Z})\ (0\leq k<|escrutinio|-1\longrightarrow_L escrutinio[k]>0)$

$Pc\equiv Pre\wedge_L i=0\wedge res=[0,0]\wedge((escrutinio[1]>escrutinio[0])\implies(fst=1\wedge snd=0))\wedge_L((escrutinio[1]\leq escrutinio[0])\implies(fst=0\wedge snd=1))$

Comentarios: Debido a la estructura de nuestro programa vamos a tener dos casos para la precondición del ciclo: Uno donde $fst=1$ y $snd=0$, y otro donde $fst=0$ y $snd=1$. Vamos a demostrar que en ambos casos se cumple la implicancia que buscamos.

Empezamos con el caso 1: $escrutinio[1]>escrutinio[0]$, es decir, $fst=1$ y $snd=0$.
Para probar que $Pc\implies I$ vemos lo que ocurre al reemplazar los valores de la Pc en el Invariante:
$Pc\implies I\ :0\leq 0\leq|escrutinio|-1\ \wedge_L\ 0\leq 0,0\leq 0\ \wedge_L\ escrutinio[1]>escrutinio[0]\ \wedge_L\ 1<|[0,0]|\ \wedge_L\ ((\forall j:\mathbb{Z})(0\leq j<0\longrightarrow_L escrutinio[0]\geq escrutinio[j])\ \wedge\ (\forall k:\mathbb{Z})(0\leq k<0\longrightarrow_L escrutinio[0]\geq escrutinio[k]))\wedge$
$(\forall p,m:\mathbb{Z})((0\leq p,m<|escrutinio|\wedge_L p\neq m)\longrightarrow_L escrutinio[p]\neq escrutinio[m])$

$Pc\implies I\ :0\leq|escrutinio|-1\wedge_L True\wedge_L escrutinio[1]>escrutinio[0]\wedge_L 1<2\wedge_L True\wedge_L True\wedge(\forall p,m:\mathbb{Z})((0\leq p,m<|escrutinio|\wedge_L p\neq m)\longrightarrow_L escrutinio[p]\neq escrutinio[m])$

$Pc\implies I\ :1\leq|escrutinio|\wedge_L escrutinio[1]>escrutinio[0]$
Y esto es verdadero, pues en Pc tenemos que $|escrutinio|>2$ y $escrutinio[1]>escrutinio[0]$ es $True$ porque $fst=1$ y $snd=0$; y respecto al predicado, este mismo se encuentra en Pc (noHayEmpate), por lo tanto:

$Pc\implies I$: True
Con lo cual probamos la implicancia del caso 1.

El caso 2, donde fst=0 y snd=1, tiene una demostración análoga:
$Pc\implies I\ :escrutinio[0]>escrutinio[1]$
Y esto es verdadero, pues en Pc tenemos que $escrutinio[0]>escrutinio[1]$ es $True$ porque $fst=0$ y $snd=1$, por lo tanto:

$Pc\implies I$: ✓ True en ambos casos

- $\{I\wedge B\}C\{I\}$

Ahora queremos ver que $\{I\wedge B\}C\{I\}$ para ello queremos demostrar que $I\wedge B\implies wp(C,I)$

$B\equiv 0\leq i<|escrutinio|-1$

$I\wedge B\equiv$
$0\leq i<|escrutinio|-1\wedge_L 0\leq snd,fst\leq i\wedge_L$
$escrutinio[fst]>escrutinio[snd]\wedge_L 1<|res|\wedge_L$
$(\forall j:\mathbb{Z})(0\leq j<i\longrightarrow_L escrutinio[fst]\geq escrutinio[j])\wedge$
$(\forall k:\mathbb{Z})(0\leq k<i\wedge k\neq fst\longrightarrow_L escrutinio[snd]\geq escrutinio[k])\wedge$
$(\forall p,m:\mathbb{Z})((0\leq p,m<|escrutinio|\wedge_L p\neq m)\longrightarrow_L escrutinio[p]\neq escrutinio[m])$

$wp(C,I)\equiv$
$((escrutinio[i]>escrutinio[fst])\wedge((I^i_{i+1})^{fst}_i)^{snd}_{fst})\vee$
$((escrutinio[i]\leq escrutinio[fst])\wedge(((escrutinio[i]>escrutinio[snd])\wedge(I^i_{i+1})^{snd}_i)\vee(escrutinio[i]\leq escrutinio[snd])\wedge(I^i_{i+1})))$

Reescribasé de otra forma:
$wp(C,I)\equiv$

$(escrutinio[i] > escrutinio[fst] \implies ((I^i_{i+1})^{fst}_i)^{snd}_{fst}) \wedge$
$(escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] > escrutinio[snd] \implies (I^i_{i+1})^{snd}_i) \wedge$
$(escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] \leq escrutinio[snd] \implies (I^i_{i+1}))$

De esta forma podremos separar el valor de verdad de la expresión dependiendo de los condicionales:

- $escrutinio[i] > escrutinio[fst]$
- $escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] > escrutinio[snd]$
- $escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] \leq escrutinio[snd]$

Caso $escrutinio[i] > escrutinio[fst]$**:**

$((I^i_{i+1})^{fst}_i)^{snd}_{fst}) \equiv$
$0 \leq i+1 \leq |escrutinio| - 1 \wedge_L 0 \leq i, fst \leq i+1 \wedge_L$
$escrutinio[i] > escrutinio[fst] \wedge_L 1 < |res| \wedge_L$
$(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[i] \geq escrutinio[j]) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[k]) \wedge$
$(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

Para este caso se puede ver que $I \wedge B \implies wp(C, I)$
Para eso verificamos los valores de verdad de los términos de $((I^i_{i+1})^{fst}_i)^{snd}_{fst})$

- $0 \leq i+1 \leq |escrutinio| - 1$. $True$. Sale de $I \wedge B$: $0 \leq i < |escrutinio| - 1$
- $0 \leq i, fst \leq i+1$. $True$. Sale de $I \wedge B$: $0 \leq fst, snd \leq i$
- $escrutinio[i] > escrutinio[fst]$. $True$ Sale de evaluar este caso precisamente
- $1 < |res|$. $True$. Sale de $I \wedge B$
- $(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[i] \geq escrutinio[j])$. $True$.
  Sale del caso que estamos evaluando y $I \wedge B$:

  $escrutinio[i] > escrutinio[fst] \wedge (\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \implies$
  $(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[i] \geq escrutinio[j])$
- $(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[k])$. $True$. Sale de $I \wedge B$:

  $(\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \implies$
  $(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[k])$
- $(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$. $True$. Sale de $I \wedge B$

Podemos concluir entonces que para este caso de i vale $I \wedge B \implies wp(C, I)$

Caso $escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] > escrutinio[snd]$**:**

$(I^i_{i+1})^{snd}_i \equiv$
$0 \leq i+1 \leq |escrutinio| - 1 \wedge_L 0 \leq fst, i \leq i+1 \wedge_L$
$escrutinio[fst] > escrutinio[i] \wedge_L 1 < |res| \wedge_L$
$(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq fst \longrightarrow_L escrutinio[i] \geq escrutinio[k]) \wedge$
$(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

Mismo análisis que en el caso anterior, se puede ver que $I \wedge B \implies wp(C, I)$
Para eso verificamos los valores de verdad de los términos de $(I^i_{i+1})^{snd}_i$

- $0 \leq i+1 \leq |escrutinio| - 1$. $True$. Sale de $I \wedge B$: $0 \leq i < |escrutinio| - 1$

- $0 \leq fst, i \leq i+1$. $True$. Sale de $I \wedge B$: $0 \leq fst, snd \leq i$

- $escrutinio[fst] > escrutinio[i]$. $True$
  Sale de evaluar la conjunción entre el caso que estamos evaluando y $I \wedge B$:

  $escrutinio[i] > escrutinio[snd] \wedge lista[fst] \geq lista[i] \wedge (\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m]) \implies$
  $escrutinio[fst] > escrutinio[i]$

- $1 < |res|$. $True$. Sale de $I \wedge B$

- $(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[j])$. $True$. Sale de $I \wedge B$

- $(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq fst \longrightarrow_L escrutinio[i] \geq escrutinio[k])$. $True$.
  Sale del caso que estamos evaluando, de uno de los items anteriores y de $I \wedge B$:

  $escrutinio[i] > escrutinio[snd] \wedge (\forall k : \mathbb{Z})(0 \leq k < i \wedge k \neq fst \longrightarrow_L escrutinio[snd] \geq escrutinio[k]) \implies$
  $(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq fst \longrightarrow_L escrutinio[i] \geq escrutinio[k])$

- $(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$. $True$. Sale de $I \wedge B$

Podemos concluir entonces que para este caso de i vale $I \wedge B \implies wp(C, I)$

Caso $escrutinio[i] \leq escrutinio[fst] \wedge escrutinio[i] \leq escrutinio[snd]$**:**

Este caso debemos analizar los valores de verdad de (I^i_{i+1}). De forma que querríamos ver que $I \wedge B \implies (I^i_{i+1})$

Comparémolos:
$I \wedge B \equiv$
$0 \leq i < |escrutinio| - 1 \wedge_L 0 \leq snd, fst \leq i \wedge_L$
$escrutinio[fst] > escrutinio[snd] \wedge_L 1 < |res| \wedge_L$
$(\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < i \wedge k \neq fst \longrightarrow_L escrutinio[snd] \geq escrutinio[k]) \wedge$
$(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

$(I^i_{i+1}) \equiv$
$0 \leq i+1 \leq |escrutinio| - 1 \wedge_L 0 \leq snd, fst \leq i+1 \wedge_L$
$escrutinio[fst] > escrutinio[snd] \wedge_L 1 < |res| \wedge_L$
$(\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq fst \longrightarrow_L escrutinio[snd] \geq escrutinio[k]) \wedge$
$(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

Veamos que valga la implicación para todos los términos

- $0 \leq i < |escrutinio| - 1 \implies 0 \leq i+1 \leq |escrutinio| - 1$
- $0 \leq snd, fst \leq i \implies 0 \leq snd, fst \leq i+1$
- $(\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \implies (\forall j : \mathbb{Z})(0 \leq j < i+1 \longrightarrow_L escrutinio[fst] \geq escrutinio[j])$
- $(\forall k : \mathbb{Z})(0 \leq k < i \wedge k \neq fst \longrightarrow_L escrutinio[snd] \geq escrutinio[k]) \implies (\forall k : \mathbb{Z})(0 \leq k < i+1 \wedge k \neq fst \longrightarrow_L escrutinio[snd] \geq escrutinio[k])$
- $(\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m]) \implies (\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

Valen todas las implicancias.
Habiendo evaluado los 3 casos que nos propusimos analizar, se puede concluir que $\{I \wedge B\}C\{I\}$

• $I \wedge \neg B \implies Qc$
Ahora probemos que $I \wedge \neg B \implies Qc$

$\neg B \equiv$ i$\geq |escrutinio| - 1$

$I \wedge \neg B \equiv \quad i = |escrutinio| - 1 \quad \wedge_L \quad 0 \leq fst, snd \leq i \quad \wedge_L \quad escrutinio[fst] > escrutinio[snd] \quad \wedge_L \quad 1 < |res| \quad \wedge_L \quad ((\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \quad \wedge \quad (\forall k : \mathbb{Z})((0 \leq k < i \wedge k \neq fst) \longrightarrow_L escrutinio[snd] \geq escrutinio[k])) \wedge (\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

I $\wedge \neg B \equiv \quad i = |escrutinio| - 1 \quad \wedge_L \quad 0 \leq fst, snd \leq |escrutinio| - 1 \quad \wedge_L \quad escrutinio[fst] > escrutinio[snd] \quad \wedge_L$
$1 < |res| \quad \wedge_L \quad ((\forall j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \quad \wedge$
$(\forall k : \mathbb{Z})((0 \leq k < |escrutinio| - 1 \wedge k \neq fst) \longrightarrow_L escrutinio[snd] \geq escrutinio[k]))$

$Qc \quad \equiv \quad 1 < |res| \quad \wedge_L \quad ((\exists i, j : \mathbb{Z})\ (0 \leq i, j < |escrutinio| - 1) \quad \wedge \quad i \neq j \quad \longrightarrow_L$
$((fst = i \leftrightarrow votosPrimero(escrutinio) = escrutinio[i]) \quad \wedge \quad (snd = j \leftrightarrow votosSegundo(escrutinio) = escrutinio[j]))$

Si recordamos cómo es el auxiliar de votosPrimero, vemos que adentro usa el predicado partidoMasVotado, el cual vemos que es exactamente como el primer predicado (con el cuantificador universal "j"), tomando a escrutinio[fst] como la variable "primero".
Lo mismo ocurre para el segundo caso, si recordamos cómo es el auxiliar de votosSegundo vemos que adentro usa el predicado segundoPartidoMasVotado, el cual es exactamente como el segundo predicado del invariante (con el cuantificador universal "k"), tomando a escrutinio[snd] como la variable "segundo".
Obviando también lo siguiente:

$1 < |res| \quad \Longrightarrow \quad 1 < |res| \quad \equiv \quad True$

Tenemos que $I \wedge \neg B \implies Qc$ ✓
Con lo que damos por demostrada la correctitud parcial del ciclo

Ahora veamos con el Teorema de Terminación si el ciclo termina y no es infinito:

- $\{I \wedge \neg B \wedge v_0 = fv\}C\{fv < v_0\}$
- $I \wedge fv \leq 0 \implies \neg B$

Proponemos una $fv = |escrutinio| - 1 - i$

Para probar $\{I \wedge \neg B \wedge v_0 = |s| - i\}C\{fv < v_0\}$, tenemos que probar que
$\{I \wedge \neg B \wedge v_0 = |escrutinio| - 1 - i\} \implies wp(C, |escrutinio| - 1 - i < v_0)$

$wp(C, |escrutinio|-1-i < v_0) \equiv wp(Sa; i := i+1, |escrutinio|-1-i < v_0) \equiv wp(Sa, wp(i := i+1, |escrutinio|-1-i < v_0))$
(En este caso llamo Sa al if grande)

$wp(i := i + 1, |escrutinio| - 1 - i < v_0) \quad \equiv \quad True \wedge_L |escrutinio| - 1 - (i + 1) < v_0$
$wp(i := i + 1, |escrutinio| - 1 - i < v_0) \quad \equiv \quad |escrutinio| - 2 - i < v_0$

Ahora dado que para en todo Sa no se realiza ninguna asignación de la variable i, al aplicar los axiomas no tendremos que hacer ningún reemplazo esta variable, por lo tanto, la parte que queremos saber de la wp ya la tenemos.

$wp(C, |escrutinio| - 1 - i < v_0) \equiv |escrutinio| - 2 - i < v_0$

De $I \wedge \neg B$ busco la parte que me interesa para probar esta implicancia:

I $\wedge \neg B \equiv$ ... $i = |escrutinio| - 1$

I $\wedge \neg B \wedge v_0 = fv \equiv$ ... $i = |escrutinio| - 1 \wedge v_0 = |escrutinio| - 1 - i$

Reemplazo en la wp a v_0: $|escrutinio| - 2 - i < v_0 \equiv |escrutinio| - 2 - i < |escrutinio| - 1 - i \equiv True$

Por lo tanto: $\{I \wedge \neg B \wedge v_0 = fv\}C\{fv < v_0\}$ ✓

Ahora probemos que $I \wedge fv \leq 0 \implies \neg B$:

I $\wedge fv \leq 0 \equiv 0 \leq i \leq |escrutinio| - 1 \wedge_L 0 \leq fst, snd \leq i \wedge_L escrutinio[fst] > escrutinio[snd] \wedge_L 1 < |res| \wedge_L ((\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \wedge (\forall k : \mathbb{Z})((0 \leq k < i \wedge k \neq fst) \longrightarrow_L escrutinio[snd] \geq escrutinio[k])) \wedge (\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m]) \wedge |escrutinio| - 1 - i \leq 0$

I $\wedge fv \leq 0 \equiv i = |escrutinio| - 1 \wedge_L 0 \leq fst, snd \leq i \wedge_L escrutinio[fst] > escrutinio[snd] \wedge_L 1 < |res| \wedge_L ((\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L escrutinio[fst] \geq escrutinio[j]) \wedge (\forall k : \mathbb{Z})((0 \leq k < i \wedge k \neq fst) \longrightarrow_L escrutinio[snd] \geq escrutinio[k])) \wedge (\forall p, m : \mathbb{Z})((0 \leq p, m < |escrutinio| \wedge_L p \neq m) \longrightarrow_L escrutinio[p] \neq escrutinio[m])$

$\neg B \equiv$ i$\geq |escrutinio| - 1$

Esta implicancia es True pues si $i = |escrutinio| - 1$ también se cumple que $|escrutinio| - 1 \geq |escrutinio| - 1$
Por lo tanto: $I \wedge fv \leq 0 \implies \neg B$ ✓

- $Pre \implies wp(S1, Pc)$

Queremos ver que la precondición implica $wp(S1, Pc)$

Definansé:
$B \equiv escrutinio[0] < escrutinio[1]$
$L1 \equiv fst = 1$
$L2 \equiv snd = 1$
y L al condicional formado por la guarda B y las asignaciones $L1$ y $L2$
Entonces tenemos que:

$wp(S1, Pc) \equiv wp(fst := 0, wp(snd := 0, wp(res := [0,0], wp(i := 0, wp(L, Pc)))))$

$wp(L, Pc) \equiv$
$2 \leq |escrutinio| \wedge$
$((escrutinio[0] < escrutinio[1] \wedge Pc_1^{fst}) \vee$
$(escrutinio[0] \geq escrutinio[1] \wedge Pc_1^{snd}))$

$wp(i = 0, wp(L, Pc)) \equiv (wp(L, Pc))_0^i$
$wp(res = [0,0], wp(i = 0, wp(L, Pc))) \equiv ((wp(L, Pc))_0^i)_{[0,0]}^{res}$
...

Así si seguimos con el resto de las asignaciones nos queda que:
$wp(S1, Pc) \equiv$
$wp(fst := 0, wp(snd := 0, wp(res := [0,0], wp(i := 0, wp(L, Pc))))) \equiv$
$((((wp(L, Pc))_0^i)_{[0,0]}^{res})_0^{snd})_0^{fst} \equiv$

$2 \leq |escrutinio| \wedge$
$((escrutinio[0] < escrutinio[1] \wedge ((((Pc_1^{fst})_0^i)_{[0,0]}^{res})_0^{snd})_0^{fst}) \vee$

$(escrutinio[0] \geq escrutinio[1] \land ((((Pc_1^{snd})_0^i)_{[0,0]}^{res})_0^{snd})_0^{fst})$

$Pc \equiv Pre \land_L i = 0 \land res = [0,0] \land ((escrutinio[1] > escrutinio[0]) \implies (fst = 1 \land snd = 0)) \land_L ((escrutinio[1] \leq escrutinio[0]) \implies (fst = 0 \land snd = 1))$

$((((Pc_1^{fst})_0^i)_{[0,0]}^{res})_0^{snd})_0^{fst} \equiv$
$Pre \land_L 0 = 0 \land [0,0] = [0,0] \land (escrutinio[1] > escrutinio[0] \implies 1 = 1 \land 0 = 0) \land_L (escrutinio[1] \leq escrutinio[0] \implies 1 = 0 \land 0 = 1) \equiv$
$Pre \land_L (escrutinio[1] \leq escrutinio[0] \implies false)$

$((((Pc_1^{snd})_0^i)_{[0,0]}^{res})_0^{snd})_0^{fst} \equiv$
$Pre \land_L 0 = 0 \land [0,0] = [0,0] \land (escrutinio[1] > escrutinio[0] \implies 0 = 1 \land 1 = 0) \land_L (escrutinio[1] \leq escrutinio[0] \implies 0 = 0 \land 1 = 1) \equiv$
$Pre \land_L (escrutinio[1] > escrutinio[0] \implies false)$

Dividamos la demostración en dos pasos:

Caso $escrutinio[1] > escrutinio[0]$
Queremos ver entonces que $Pre \implies wp(S1, Pc)$ que en este caso equivale a
$2 \leq |escrutinio| \land (escrutinio[0] < escrutinio[1] \land Pre \land_L (escrutinio[1] \leq escrutinio[0] \implies false))$

Ahora, por el caso que estamos analizando, ello es equivalente a:
$2 \leq |escrutinio| \land Pre$
Se puede ver que $Pre \implies 2 \leq |escrutinio| \land Pre$ pues uno de los términos de Pre es $2 < |escrutinio|$, por lo que concluímos la demostración para este caso

Caso $escrutinio[1] \leq escrutinio[0]$
Análogo al caso anterior queremos ver entonces que $Pre \implies wp(S1, Pc)$ que en este caso equivale a
$2 \leq |escrutinio| \land (escrutinio[0] \leq escrutinio[1] \land Pre \land_L (escrutinio[1] > escrutinio[0] \implies false))$

Ahora, por el caso que estamos analizando, ello es equivalente a:
$2 \leq |escrutinio| \land Pre$
Se puede ver que $Pre \implies 2 \leq |escrutinio| \land Pre$ pues uno de los términos de Pre es $2 < |escrutinio|$, por lo que concluímos la demostración para este caso

Finalmente, probamos que $Pre \implies wp(S1, Pc)$ ✓

Al probar los dos items del Teorema de Terminación, probamos que el ciclo termina.
Finalmente al probar que se cumplen el Teorema del Invariante y el Teorema de Terminación, podemos concluir que
$Pc \implies wp(C, Qc) \equiv \{Pc\}C\{Qc\}$ ✓ ; con lo cual sumado a que
$Pre \implies wp(S1, Pc)$ ✓ y
$Qc \implies wp(S2, Post)$ ✓
podemos decir que se cumple la tripla de Hoare $\{Pre\}S\{Post\}$, es decir, que finalemente el programa es correcto respecto a su especificación.

Rexona®

NO TE ABANDONA