

Derivación de instrucciones simples

$$\{ A \equiv m = M \wedge n = N \wedge XY = u + mn \}$$

instrucciones a derivar

$$\{ B \equiv m = M \text{ div } 2 \wedge n = 2N \wedge XY = u + mn \}$$

$$\langle m, n \rangle := \langle m \text{ div } 2, 2 * n \rangle$$

$$?A \Rightarrow pmd(\langle m, n \rangle := \langle m \text{ div } 2, 2 * n \rangle, B)?$$

$$B_{m,n}^{m \text{ div } 2, 2*n} \Leftrightarrow m \text{ div } 2 = M \text{ div } 2 \wedge 2n = 2N \wedge XY = u + (m \text{ div } 2)2n$$

$$m \text{ es par } mn = (m \text{ div } 2)2n \text{ y } A \wedge \text{par}(m) \Rightarrow B_{m,n}^{m \text{ div } 2, 2*n}$$

$$m \text{ no es par } mn = (m \text{ div } 2)2n + n, \text{ luego } XY = u + n + (m \text{ div } 2)2n \text{ y}$$

$$A \wedge \neg \text{par}(m) \Rightarrow B_{m,n,u}^{m \text{ div } 2, 2*n, u+n}$$

$$\{ A \equiv m = M \wedge n = N \wedge XY = u + mn \}$$

$$\text{si } \text{par}(m) \text{ entonces } \langle m, n \rangle := \langle m \text{ div } 2, 2 * n \rangle$$

$$\text{si no } \langle m, n, u \rangle := \langle m \text{ div } 2, 2 * n, u + n \rangle \text{ fsi}$$

$$\{ B \equiv m = M \text{ div } 2 \wedge n = 2N \wedge XY = u + mn \}$$

Esquema de derivación de bucles

```

{ Pre.  $A$  }
   $P_0$  ;           {inicialización}
{ Inv.  $I$ , Cota  $C$  }
  mientras  $b$  hacer
    {  $I \wedge b$  }
     $P_1$  ;         {restablecer}
    {  $R$  }
     $P_2$            {avanzar}
    {  $I$  }
  fmientras
{ Post.  $B$  }

```

- ① Diseñar I y b a partir de B tal que $I \wedge \neg b \Rightarrow B$.
- ② Diseñar P_0 tal que $\{ A \} P_0 \{ I \}$.
- ③ Diseñar C tal que $I \wedge b \Rightarrow C \geq 0$.
- ④ Diseñar P_2 y construir $R \equiv \text{pmd}(P_2, I)$.
- ⑤ Diseñar P_1 comparando $I \wedge b$ con R y tal que $\{ I \wedge b \} P_1 \{ R \}$.
- ⑥ Comprobar $\{ I \wedge b \wedge C = z \} P_1 ; P_2 \{ C < z \}$.

División entera

```

{  $A \equiv m \geq 0 \wedge n > 0$  }
fun div-ent( $m, n : ent$ ) dev  $\langle q, r : ent \rangle$ 
{  $B \equiv m = n * q + r \wedge 0 \leq r \wedge r < n$  }

```

Postcondición **conjuntiva** $R_1 \wedge R_2$: una parte como **invariante** y la otra como **negación de la condición del bucle**.

$$\neg b \equiv m = n * q + r \quad I \equiv 0 \leq r \wedge r < n$$

$$\neg b \equiv 0 \leq r \quad I \equiv m = n * q + r \wedge r < n$$

$$\neg b \equiv r < n \quad I \equiv m = n * q + r \wedge 0 \leq r$$

División entera

Inicialización $\{ A \} \langle q, r \rangle := \langle 0, m \rangle \{ I \}?$

$$\begin{aligned} (m = n * q + r \wedge 0 \leq r)_{q,r}^{0,m} &\Leftrightarrow m = n * 0 + m \wedge 0 \leq m \\ &\Leftarrow m \geq 0 \wedge n > 0 \end{aligned}$$

Función de cota $C = r \geq 0$

Avanzar $r := r - 1$

¿Podemos avanzar más rápido? $r \geq n \longrightarrow r := r - n$

$$\begin{aligned} R \equiv I_r^{r-n} &\Leftrightarrow m = n * q + (r - n) \wedge 0 \leq (r - n) \\ &\Leftrightarrow m = n * (q - 1) + r \wedge 0 \leq (r - n) \stackrel{?}{\Leftarrow} I \wedge b \end{aligned}$$

Restablecer $\{ I \wedge b \} q := q + 1 \{ R \}?$

$$R_q^{q+1} \Leftrightarrow m = n * (q + 1) + (r - n) \wedge 0 \leq (r - n) \Leftarrow I \wedge b$$

Terminación $\{ I \wedge b \wedge r = z \} q := q + 1 ; r := r - n \{ r < z \}?$

$$((r < z)_r^{r-n})_q^{q+1} \Leftrightarrow r - n < z \stackrel{?}{\Leftarrow} I \wedge b \wedge r = z$$

Debemos añadir $n > 0$ al invariante.

División entera

$\{m \geq 0 \wedge n > 0\}$
 $\langle q, r \rangle := \langle 0, m \rangle ;$
 $\{m = n * q + r \wedge r \geq 0 \wedge n > 0\}$
mientras $r \geq n$ **hacer**
 $q := q + 1 ;$
 $r := r - n$
fmientras
 $\{m = n * q + r \wedge 0 \leq r \wedge r < n\}$

Coste: $\Theta(m \operatorname{div} n)$

Potencia

```
{  $A \equiv m > 0 \wedge n \geq 0$  }  
fun potencia( $m, n : ent$ ) dev  $r : ent$   
{  $B \equiv r = m^n$  }
```

No hay conjunciones. Sustituir constantes (parámetros de entrada) por nuevas variables:

- $r = m^x \wedge x = n$
- $r = y^n \wedge y = m$
- $r = y^x \wedge y = m \wedge x = n$

$$\neg b \equiv r = m^x \quad I \equiv x = n$$

$$\neg b \equiv x = n \quad I \equiv r = m^x$$

Potencia

Inicialización $\langle x, r \rangle := \langle 0, 1 \rangle$

$$(r = m^x)_{x,r}^{0,1} \Leftrightarrow 1 = m^0 \Leftrightarrow \text{cierto}$$

Función de cota $n - x$. Añadimos al invariante $0 \leq x \leq n$:

$$I \wedge b \Rightarrow n - x \geq 0$$

Avanzar $x := x + 1$

$$R \equiv I_x^{x+1} \Leftrightarrow r = m^{x+1} \wedge 0 \leq x + 1 \leq n \stackrel{?}{\Leftarrow} I \wedge b$$

Restablecer $r := m * r$

Terminación $\{I \wedge b \wedge n - x = z\} r := m * r; x := x + 1 \{n - x < z\}$

Potencia

```
{  $m > 0 \wedge n \geq 0$  }  
fun potencia( $m, n : ent$ ) dev  $r : ent$   
var  $x : ent$   
   $\langle x, r \rangle := \langle 0, 1 \rangle ;$   
  {  $I \equiv 0 \leq x \leq n \wedge r = m^x$  }  
  mientras ( $x \neq n$ ) hacer  
     $r := r * m ;$   
     $x := x + 1$   
  fmientras  
ffun  
{  $r = m^n$  }
```

Coste: $\Theta(n)$

Ejercicio

Desarrollar las otras alternativas.

Suma de elementos buenos

$\{N \geq 1\}$

fun suma-buenos($X[0..N]$ **de** ent) **dev** $s : ent$

$\{s = (\sum i : 0 \leq i < N \wedge \text{bueno}(i, X) : X[i])\}$

$\text{bueno}(i, X) \equiv (X[i] = 2^i)$

No utilizar ninguna operación que calcule potencias.

$$I \equiv s = (\sum i : 0 \leq i < n \wedge \text{bueno}(i, X) : X[i]) \wedge 0 \leq n \leq N$$

$$b \equiv n \neq N$$

Inicialización $\langle n, s \rangle := \langle 0, 0 \rangle$

Función de cota $N - n$

Avanzar $n := n + 1$

Suma de elementos buenos

$$I_n^{n+1} \equiv s = (\sum i : 0 \leq i < n+1 \wedge \text{bueno}(i, X) : X[i]) \wedge 0 \leq n+1 \leq N$$

$$\stackrel{?}{\Leftarrow} I \wedge b$$

La última parte:

$$0 \leq n \leq N \wedge n \neq N \Rightarrow 0 \leq n+1 \leq N.$$

Para hacer cierta la primera igualdad:

$$s = (\sum i : 0 \leq i < n+1 \wedge \text{bueno}(i, X) : X[i])$$

$$\Leftrightarrow s = (\sum i : 0 \leq i < n \wedge \text{bueno}(i, X) : X[i]) + \begin{cases} X[n] & \text{si } \text{bueno}(n, X) \\ 0 & \text{si } \neg \text{bueno}(n, X) \end{cases}$$

Suma de elementos buenos

```

 $\langle n, s \rangle := \langle 0, 0 \rangle ;$ 
 $\{I \equiv s = (\sum i : 0 \leq i < n \wedge \text{bueno}(i, X) : X[i]) \wedge 0 \leq n \leq N\}$ 
mientras  $n \neq N$  hacer
     $\{I \wedge n \neq N\}$ 
    si  $\text{bueno}(n, X)$  entonces  $s := s + X[n]$  fsi ;
     $\{I_n^{n+1}\}$ 
     $n := n + 1$ 
fmientras
  
```

¿Cómo comprobar **eficientemente** $\text{bueno}(n, X)$?

Introducir en el invariante una nueva variable $p = 2^n$:

$$\text{bueno}(n, X) \Leftrightarrow X[n] = p$$

Inicialización de la nueva variable: $p := 1$.

Suma de elementos buenos

$\langle n, s, p \rangle := \langle 0, 0, 1 \rangle ;$

$\{I \wedge p = 2^n\}$

mientras $n \neq N$ **hacer**

$\{I \wedge p = 2^n \wedge n \neq N\}$

si $X[n] = p$ **entonces** $s := s + X[n]$ **fsi** ;

$\{I_n^{n+1} \wedge p = 2^n\}$

~~restablecer~~ $p??$

$\{I_n^{n+1} \wedge p = 2^{n+1}\}$

$n := n + 1$

fmientras

$$p = 2^{n+1} \Leftrightarrow p = 2 * 2^n.$$

Suma de elementos buenos

```

{N ≥ 1}
fun suma-buenos(X[0..N] de ent) dev s : ent
var p, n : ent
    ⟨n, s, p⟩ := ⟨0, 0, 1⟩ ;
    {I ∧ p = 2n}
    mientras n ≠ N hacer
        si X[n] = p entonces s := s + X[n] fsi ;
        p := 2 * p ;
        n := n + 1
    fmientras
ffun
{s = (∑ i : 0 ≤ i < N ∧ bueno(i, X) : X[i])}
  
```

Coste: $\Theta(N)$

Sustituir N por n permite realizar un **recorrido del vector de izquierda a derecha**.

Ejercicio

Probar sustituyendo 0 con n .

Segmento de suma máxima

Dado un vector no vacío de enteros, calcular la suma del segmento no vacío de suma máxima.

Un par p, q representa el segmento $[p, q)$.

$\{N \geq 1\}$

fun seg-suma-máx($X[0..N)$ **de** ent) **dev** $r : ent$
 $\{r = (\text{máx } p, q : 0 \leq p < q \leq N : \mathcal{S}(p, q))\}$

$\mathcal{S}(p, q) = (\sum i : p \leq i < q : X[i]).$

$$I \equiv 1 \leq n \leq N \wedge r = (\text{máx } p, q : 0 \leq p < q \leq n : \mathcal{S}(p, q))$$

$$b \equiv n \neq N$$

Segmento de suma máxima

Inicialización $\langle n, r \rangle := \langle 1, X[0] \rangle$

Función de cota $N - n$

Avanzar $n := n + 1$

$$1 \leq n \leq N \wedge n \neq N \Rightarrow 1 \leq n + 1 \leq N.$$

$$\begin{aligned} & (\text{máx } p, q : 0 \leq p < q \leq n + 1 : \mathcal{S}(p, q)) \\ = & (\text{máx } p, q : 0 \leq p < q \leq n : \mathcal{S}(p, q)) \\ & \text{máx} \\ & (\text{máx } p : 0 \leq p < n + 1 : \mathcal{S}(p, n + 1)) \\ \stackrel{I}{=} & r \text{ máx } (\text{máx } p : 0 \leq p < n + 1 : \mathcal{S}(p, n + 1)). \end{aligned}$$

Añadimos al invariante esta expresión, pero con n en vez de $n + 1$ pues $\mathcal{S}(p, N + 1)$ no está bien definido:

$$S \equiv s = (\text{máx } p : 0 \leq p < n : \mathcal{S}(p, n)).$$

Cuando $n = 1$, se tiene $s = \mathcal{S}(0, 1) = X[0]$.

Segmento de suma máxima

```

var  $r, n, s : \text{ent}$ 
 $\langle n, r, s \rangle := \langle 1, X[0], X[0] \rangle ;$ 
mientras  $n \neq N$  hacer
     $\{I \wedge S \wedge n \neq N\}$ 
    restablecer  $s??$ 
     $\{I \wedge S_n^{n+1}\}$ 
     $r := r \text{ máx } s ;$ 
     $\{I_n^{n+1} \wedge S_n^{n+1}\}$ 
     $n := n + 1$ 
     $\{I \wedge S\}$ 
fmientras
  
```

Desarrollando la expresión del máximo en S_n^{n+1} :

$$\begin{aligned}
 & (\text{máx } p : 0 \leq p < n + 1 : \mathcal{S}(p, n + 1)) \\
 = & (\text{máx } p : 0 \leq p < n : \mathcal{S}(p, n + 1)) \text{ máx } \mathcal{S}(n, n + 1) \\
 = & (\text{máx } p : 0 \leq p < n : (\mathcal{S}(p, n) + X[n])) \text{ máx } X[n] \\
 = & ((\text{máx } p : 0 \leq p < n : \mathcal{S}(p, n)) + X[n]) \text{ máx } X[n] \\
 \stackrel{I}{=} & (s + X[n]) \text{ máx } X[n]
 \end{aligned}$$

Segmento de suma máxima

```
{N ≥ 1}
fun seg-suma-máx(X[0..N] de ent) dev r : ent
  var r, n, s : ent
  ⟨ n, r, s ⟩ := ⟨ 1, X[0], X[0] ⟩ ;
  {I ∧ S}
  mientras n ≠ N hacer
    s := (s + X[n]) máx X[n] ;
    r := r máx s ;
    n := n + 1
  fmientras
ffun
  {r = (máx p, q : 0 ≤ p < q ≤ N : S(p, q))}
```

Coste: $\Theta(N)$.

Segmento de suma máxima (2)

Devolver el segmento correspondiente (además de la suma).

$\{N \geq 1\}$

fun seg-suma-máx($X[0..N]$ **de** ent) **dev** $\langle r : ent, a, b : 0..N \rangle$
 $\{r = (\text{máx } p, q : 0 \leq p < q \leq N : \mathcal{S}(p, q)) \wedge 0 \leq a < b \leq N \wedge r = \mathcal{S}(a, b)\}$

Enriquecemos el invariante:

$$I \equiv 1 \leq n \leq N \wedge r = (\text{máx } p, q : 0 \leq p < q \leq n : \mathcal{S}(p, q)) \\ \wedge 0 \leq a < b \leq n \wedge r = \mathcal{S}(a, b)$$

Añadimos la inicialización: $\langle a, b \rangle := \langle 0, 1 \rangle$.

¿Cómo afecta la actualización de r a a y b ?

Variable c : misma información con respecto a s que a, b con respecto a r .

Añadimos al invariante

$$R \equiv s = \mathcal{S}(c, n) \wedge 0 \leq c < n$$

Inicialización: $c := 0$.

Segmento de suma máxima (2)

¿Cómo restablecer a y b ?

```
mientras  $n \neq N$  hacer  
   $\{I \wedge S \wedge R \wedge n \neq N\}$   
   $s := (s + X[n]) \text{ máx } X[n];$   
  restablecer  $c??$   
   $\{I \wedge S_n^{n+1} \wedge R_n^{n+1}\}$   
  si  $r < s$  entonces  $\langle r, a, b \rangle := \langle s, c, n + 1 \rangle$  fsi ;  
   $n := n + 1$   
fmientras
```

Segmento de suma máxima (2)

```

{N ≥ 1}
fun seg-suma-máx(X[0..N] de ent) dev ⟨ r : ent, a, b : nat ⟩
  var n, s, c : ent
  ⟨ n, r, a, b, s, c ⟩ := ⟨ 1, X[0], 0, 1, X[0], 0 ⟩ ;
  mientras n ≠ N hacer
    {I ∧ S ∧ R ∧ n ≠ N}
    si s ≥ 0 entonces s := s + X[n]
      si no ⟨ s, c ⟩ := ⟨ X[n], n ⟩
    fsi ;
    {I ∧ Snn+1 ∧ Rnn+1}
    si r < s entonces ⟨ r, a, b ⟩ := ⟨ s, c, n + 1 ⟩ fsi ;
    n := n + 1
  fmientras
  {r = (máx p, q : 0 ≤ p < q ≤ N : S(p, q)) ∧ 0 ≤ a < b ≤ N ∧ r = S(a, b)}

```

Coste: $\Theta(N)$.

Problemas de búsqueda

Raíz cuadrada entera de n : “buscar el mayor natural i tal que $i^2 \leq n$ ”

$$r = (\text{máx } i : 0 \leq i \wedge i^2 \leq n : i)$$

O también:

$$r = (\text{mín } i : 0 \leq i \wedge (i+1)^2 > n : i)$$

es decir, buscamos el menor natural i tal que $(i+1)^2 > n$.

- Búsqueda lineal
- Búsqueda lineal acotada
- Búsqueda binaria

Búsqueda lineal (de menor a mayor)

Encontrar el **primer elemento que cumpla una propiedad $P(i)$** a partir de una cota inferior c_{inf} , y sabemos que existe.

```

{  $(\exists i : i \geq c_{inf} : P(i))$  }
fun búsqueda-lineal(...) dev  $x : ent$ 
{  $x = (\text{mín } i : i \geq c_{inf} \wedge P(i) : i)$  }
  
```

Otra forma de expresar la postcondición es

$$x \geq c_{inf} \wedge P(x) \wedge (\forall i : c_{inf} \leq i < x : \neg P(i))$$

Invariante $I \equiv x \geq c_{inf} \wedge (\forall i : c_{inf} \leq i < x : \neg P(i))$

Condición del bucle $\neg b \equiv P(x)$

Inicialización $x := c_{inf}$

Avanzar $x := x + 1$

$$I \wedge b \Rightarrow I_x^{x+1} \text{ no hace falta restablecer}$$

Búsqueda lineal (de menor a mayor)

```

{  $(\exists i : i \geq c_{inf} : P(i))$  }
fun búsqueda-lineal(...) dev  $x : ent$ 
   $x := c_{inf}$  ;
  mientras  $\neg P(x)$  hacer
     $x := x + 1$ 
  fmientras
ffun
{  $x = (\text{mín } i : i \geq c_{inf} \wedge P(i) : i)$  }

```

Terminación

$$I \Rightarrow (\forall i : i \geq c_{inf} \wedge P(i) : i \geq x)$$

Por la precondition, existe un M tal que $M \geq c_{inf} \wedge P(M)$.

$$I \Rightarrow M \geq x \Leftrightarrow \underbrace{M - x}_{\text{cota}} \geq 0$$

Búsqueda lineal acotada

Dado un vector de booleanos $B[0..N)$, con $N \geq 0$, devolver en x el menor valor i , con $0 \leq i < N$, tal que $B[i]$ sea cierto (ÉXITO).

Si no existe, x debe valer N (FALLO).

$\{ N \geq 0 \}$

fun búsqueda-lineal-acotada($B[0..N)$ **de** *bool*) **dev** $x : \text{ent}$

$\{ (x = N \wedge (\forall i : 0 \leq i < N : \neg B[i])) \vee_c x = (\text{mín } i : 0 \leq i < N \wedge B[i] : i) \}$

Reescribir la postcondición como

$$0 \leq x \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i]) \wedge P(x)$$

$$P(x) = (0 \leq x < N \wedge_c B[x]) \vee (x = N)$$

N es el centinela; $B[N]$ no puede ser accedido

Búsqueda lineal acotada

Invariante $I \equiv 0 \leq x \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i])$.

Condición del bucle $\neg P(x) \Leftrightarrow x \neq N \wedge_c \neg B[x]$.

Inicialización $x := 0$

Función de cota $N - x$

Avanzar $x := x + 1$

$$\begin{aligned} I_x^{x+1} &\Leftrightarrow 0 \leq x + 1 \leq N \wedge (\forall i : 0 \leq i < x + 1 : \neg B[i]) \\ &\Leftarrow I \wedge (x \neq N \wedge_c \neg B[x]) \end{aligned}$$

$\{ N \geq 0 \}$

$x := 0 ;$

$\{ I \equiv 0 \leq x \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i]) \}$

mientras $(x \neq N \wedge_c \neg B[x])$ **hacer**

$x := x + 1$

fmientras

$\{ (x = N \wedge (\forall i : 0 \leq i < N : \neg B[i])) \vee_c x = (\text{mín } i : 0 \leq i < N \wedge B[i] : i) \}$

Búsqueda lineal acotada: otra posibilidad

Invariante $I \equiv 0 \leq x \leq y \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i]) \wedge P(y)$

Condición del bucle $x \neq y$

Inicialización $\langle x, y \rangle := \langle 0, N \rangle$

Función de cota $y - x$

Avanzar $x := x + 1$

$$I_x^{x+1} \Leftrightarrow 0 \leq x + 1 \leq y \leq N \wedge (\forall i : 0 \leq i < x + 1 : \neg B[i]) \wedge P(y)$$

$$\Leftrightarrow 0 \leq x + 1 \leq y \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i]) \wedge \neg B[x] \wedge P(y)$$

$$\stackrel{?}{\Leftarrow} I \wedge x \neq y$$

Solo si $\neg B[x]$

Restablecer **si** $\neg B[x]$ **entonces** $x := x + 1$ **si no** $y := x$ **fsi**

$\{ N \geq 0 \}$

$\langle x, y \rangle := \langle 0, N \rangle ;$

$\{ I \equiv 0 \leq x \leq y \leq N \wedge (\forall i : 0 \leq i < x : \neg B[i]) \wedge P(y) \}$

mientras $(x \neq y)$ **hacer**

si $\neg B[x]$ **entonces** $x := x + 1$ **si no** $y := x$ **fsi**

fmientras

$\{ (x = N \wedge (\forall i : 0 \leq i < N : \neg B[i])) \vee_c x = (\text{mín } i : 0 \leq i < N \wedge B[i] : i) \}$

Ejemplo

En un vector de enteros un índice es *gordote* si el valor del vector en dicha posición es igual a la suma de los valores de todas las posiciones que le siguen. Determinar el mayor índice *gordote* de un vector de enteros $V[1..N]$, con $N \geq 0$. En caso de no existir ningún índice *gordote* el resultado debe ser 0.

$\{ N \geq 0 \}$

fun gordote($V[1..N]$ **de** ent) **dev** $x : nat$

$\{ (x = 0 \wedge (\forall i : 1 \leq i \leq N : \neg \text{gordote}(V, i)))$

$\vee_c x = (\text{máx } i : 1 \leq i \leq N \wedge \text{gordote}(V, i) : i) \}$

$\text{gordote}(V, j) \equiv (V[j] = (\sum k : j + 1 \leq k \leq N : V[k]))$

Simétrico al esquema de búsqueda lineal acotada:

$P(x) \equiv (x = 0) \vee (0 < x \leq N \wedge_c \text{gordote}(V, x))$

$\langle x, y \rangle := \langle N, 0 \rangle ;$

$\{ I \equiv (\forall i : x + 1 \leq j \leq N : \neg \text{gordote}(V, i)) \wedge P(y) \wedge 0 \leq y \leq x \leq N \}$

mientras $(x \neq y)$ **hacer**

si $\neg \text{gordote}(V, x)$ **entonces** $x := x - 1$ **si no** $y := x$ **fsi**

fmientras

¿Comprobación eficiente de $\text{gordote}(V, x)$?

Introducir una variable $S \equiv s = (\sum k : x + 1 \leq k \leq N : V[k])$.

Comprobar $V[x] = s$.

Inicialización $s := 0$

Restablecer $I \wedge S \wedge (x \neq y) \wedge (V[x] \neq s) \Rightarrow ((I \wedge S)_x^{x-1})_s^{s+V[x]}$

$\{ N \geq 0 \}$

fun $\text{gordote}(V[1..N] \text{ de } ent) \text{ dev } x : nat$

var $y, s : ent$

$\langle x, y, s \rangle := \langle N, 0, 0 \rangle ;$

$\{ I \wedge S \}$

mientras $(x \neq y)$ **hacer**

si $V[x] \neq s$ **entonces** $s + V[x] ; x := x - 1$ **si no** $y := x$ **fsi**

fmientras

ffun

$\{ (x = 0 \wedge (\forall i : 1 \leq i \leq N : \neg \text{gordote}(V, i)))$

$\vee_c x = (\text{máx } i : 1 \leq i \leq N \wedge \text{gordote}(V, i) : i) \}$

Coste: $\Theta(N)$

Búsqueda binaria

```

{  $N \geq 1 \wedge f(0) \leq A < f(N)$  }
fun búsqueda-binaria( $A : ent...$ ) dev  $x : ent$ 
{  $f(x) \leq A < f(x+1) \wedge 0 \leq x < N$  }

```

Introducimos una nueva variable: $f(x) \leq A < f(y) \wedge y = x + 1$

Invariante $I \equiv f(x) \leq A < f(y) \wedge 0 \leq x < N \wedge x < y \leq N$

Condición del bucle $b \equiv y \neq x + 1$ **hay un valor entre x e y**

Inicialización $\langle x, y \rangle := \langle 0, N \rangle$

Función de cota $y - x$

Avanzar Elegimos h tal que $x < h < y$

$$I_x^h \Leftrightarrow f(h) \leq A < f(y) \wedge 0 \leq h < N \wedge h < y \leq N \stackrel{?}{\Leftarrow} I \wedge b$$

$$I_y^h \Leftrightarrow f(x) \leq A < f(h) \wedge 0 \leq x < N \wedge x < h \leq N \stackrel{?}{\Leftarrow} I \wedge b$$

casos

$$\begin{aligned}
 & f(h) \leq A \rightarrow x := h \\
 & \square f(h) > A \rightarrow y := h
 \end{aligned}$$

fcasos

¿Cuál es el mejor h ? $h := (x + y) \text{ div } 2$

Búsqueda binaria

```

{  $N \geq 1 \wedge f(0) \leq A < f(N)$  }
fun búsqueda-binaria( $A : ent...$ ) dev  $x : ent$ 
var  $y, h : ent$ 
     $\langle x, y \rangle := \langle 0, N \rangle ;$ 
    {  $I \equiv f(x) \leq A < f(y) \wedge 0 \leq x < N \wedge x < y \leq N$  }
    mientras  $y \neq x + 1$  hacer
         $h := (x + y) \text{ div } 2 ;$ 
        casos
             $f(h) \leq A \rightarrow x := h$ 
             $\square f(h) > A \rightarrow y := h$ 
        fcasos
    fmientras
ffun
    {  $f(x) \leq A < f(x + 1) \wedge 0 \leq x < N$  }
  
```

La distancia entre x e y se reduce a la mitad en cada vuelta.

Coste: $\Theta(\log N)$.

Ejemplo: raíz cuadrada entera

```

{  $n \geq 0$  }
fun raíz-ent-log( $n : \text{ent}$ ) dev  $r : \text{ent}$ 
{  $r \geq 0 \wedge r^2 \leq n < (r+1)^2$  }

```

$$f(x) = x^2$$

$$n \geq 0 \Rightarrow 0^2 \leq n < (n+1)^2$$

```

fun raíz-ent-log( $n : \text{ent}$ ) dev  $r : \text{ent}$     {  $\Theta(\log n)$  }
var  $y, h : \text{ent}$ 
     $\langle r, y \rangle := \langle 0, n+1 \rangle$  ;
    mientras  $y \neq r+1$  hacer
         $h := (r+y) \text{ div } 2$  ;
        si  $h * h \leq n$  entonces  $r := h$ 
        si no  $y := h$ 
    fsi
    fmientras
ffun

```

Buscar un elemento en un vector ordenado

En el algoritmo búsqueda-binaria $0 < h < N \Rightarrow f(0)$ y $f(N)$ no se consultan

La precondition se utiliza solo para inicializar x e y .

Relajamos en la precondition

$$f(0) \leq A < f(N) \vee f(0) > A \vee f(N) \leq A$$

Y la postcondición sería

$$0 \leq x < N \wedge (f(x) \leq A < f(x+1) \vee f(0) > A \vee f(N) \leq A)$$

Buscar un elemento en un vector ordenado

```
{  $N \geq 1 \wedge (\forall i, j : 0 \leq i \leq j < N : V[i] \leq V[j])$  }
fun está?( $V[0..N]$  de  $ent, A : ent$ ) dev  $r : bool$ 
{  $r = (\exists i : 0 \leq i < N : V[i] = A)$  }
```

Buscar un elemento en un vector ordenado

$$f(x) = \begin{cases} V[x] & \text{si } 0 \leq x < N \\ \infty & \text{e.o.c.} \end{cases}$$

La postcondición se simplifica:

$$0 \leq x < N \wedge (f(x) \leq A < f(x+1) \vee f(0) > A)$$

Si se cumple la postcondición y además **el vector está ordenado**:

$$(\exists i : 0 \leq i < N : V[i] = A) \Leftrightarrow (V[x] = A)$$

```

fun está?( $V[0..N]$  de  $ent, A : ent$ ) dev  $r : bool$     {  $\Theta(\log n)$  }
var  $x, y, h : ent$ 
     $\langle x, y \rangle := \langle 0, N \rangle ;$ 
    mientras  $y \neq x + 1$  hacer
         $h := (x + y) \text{ div } 2 ;$ 
        casos
             $V[h] \leq A \rightarrow x := h$ 
             $\square V[h] > A \rightarrow y := h$ 
        fcasos
    fmientras ;
     $r := (V[x] = A)$ 
ffun
  
```

Manipulación de vectores mediante intercambios

Muchos problemas se resuelven intercambiando posiciones de un vector.

$$\left. \begin{array}{l} aux := v[i] ; \\ v[i] := v[j] ; \\ v[j] := aux \end{array} \right\} \text{intercambiar}(v, i, j)$$

Extendemos la notación de $\text{asig}(v, x, y, X, Y)[i] = \begin{cases} v[i] & \text{si } i \neq x \wedge i \neq y \\ X & \text{si } i = x \\ Y & \text{si } i = y \end{cases}$

$$\frac{P \Rightarrow \text{def}(\text{asig}(v, i, j, v[j], v[i])) \wedge Q_v^{\text{asig}(v, i, j, v[j], v[i])}}{\{P\} \text{intercambiar}(v, i, j) \{Q\}}$$

La bandera holandesa



$$\{ N \geq 0 \wedge v = V \}$$

proc bandera(**E/S** $v[0..N]$ **de** $\{A, B, R\}$)

$$\{ v \in \text{Perm}(V) \wedge (\exists p, q : 0 \leq p \leq q \leq N : (\forall i : 0 \leq i < p : v[i] = R) \\ \wedge (\forall j : p \leq j < q : v[j] = B) \\ \wedge (\forall k : q \leq k < N : v[k] = A))) \}$$

Solo se permiten operaciones de **intercambio** en el vector.

Invariante $I \equiv v \in \text{Perm}(V) \wedge P_r \wedge P_b \wedge P_a \wedge 0 \leq r \leq b \leq a \leq N$

$$P_r \equiv (\forall i : 0 \leq i < r : v[i] = R)$$

$$P_b \equiv (\forall j : r \leq j < b : v[j] = B)$$

$$P_a \equiv (\forall k : a \leq k < N : v[k] = A)$$

Entre b y $a - 1$ aún no se han procesado los elementos del vector.

Condición de terminación del bucle $b = a$

Inicialización $\langle r, b, a \rangle := \langle 0, 0, N \rangle$

Función de cota $a - b$

La bandera holandesa

Avanzar consultamos $v[b]$:

$\langle r, b, a \rangle := \langle 0, 0, N \rangle ;$

mientras $(b \neq a)$ **hacer**

casos

$v[b] = R \rightarrow S_r$

$\square v[b] = B \rightarrow S_b$

$\square v[b] = A \rightarrow S_a$

fcasos

fmientras

S_b avanzar b : $I \wedge v[b] = B \Rightarrow I_b^{b+1}$

S_a intercambiar $v[b]$ y $v[a-1]$ para colocar un azul más:

$\{I \wedge (b \neq a) \wedge (v[b] = A)\}$

intercambiar($v, b, a-1$) ;

$\{v \in \text{Perm}(V) \wedge P_r \wedge P_b \wedge P_a \wedge 0 \leq r \leq b < a \leq N \wedge (v[a-1] = A)\}$

$a := a - 1$

$\{I\}$

La bandera holandesa

S_r a partir de P_b sabemos que:

- $r < b \Rightarrow v[r] = B$

Intercambiando $v[r]$ con $v[b]$ colocaremos un rojo y un blanco más:

$$\{I \wedge (b \neq a) \wedge (v[b] = R) \wedge (r < b) \wedge (v[r] = B)\}$$

intercambiar(v, b, r);

$$\{v \in \text{Perm}(V) \wedge P_r \wedge (\forall i : r + 1 \leq i < b : v[i] = B) \wedge P_a \wedge \\ (v[r] = R) \wedge 0 \leq r < b < a \leq N \wedge (v[b] = B)\}$$

$$\langle r, b \rangle := \langle r + 1, b + 1 \rangle$$

$$\{I\}$$

- $r = b \Rightarrow$ **no hay blancos colocados de momento**

Tendremos solamente colocado un rojo más.

Un intercambio entre $v[r]$ con $v[b]$ no afecta al vector y evitamos una distinción de casos:

$$\{I \wedge (b \neq a) \wedge (v[b] = R) \wedge (r = b)\}$$

intercambiar(v, b, r)

$$\{v \in \text{Perm}(V) \wedge P_r \wedge P_a \wedge 0 \leq r = b < a \leq N \wedge (v[r] = R)\}$$

$$\langle r, b \rangle := \langle r + 1, b + 1 \rangle$$

$$\{I\}$$

La bandera holandesa

```

{  $N \geq 0 \wedge v = V$  }
proc bandera(E/S  $v[0..N]$  de  $\{A, B, R\}$ )
var  $r, b, a : \text{ent}$ 
     $\langle r, b, a \rangle := \langle 0, 0, N \rangle ;$ 
    {  $I \equiv P_r \wedge P_b \wedge P_a \wedge 0 \leq r \leq b \leq a \leq N$  }
    mientras  $(b \neq a)$  hacer
        casos
             $v[b] = R \rightarrow \text{intercambiar}(v, b, r) ; \langle r, b \rangle := \langle r + 1, b + 1 \rangle$ 
             $\square v[b] = B \rightarrow b := b + 1$ 
             $\square v[b] = A \rightarrow \text{intercambiar}(v, b, a - 1) ; a := a - 1$ 
        fcasos
    fmientras
ffun
    {  $v \in \text{Perm}(V) \wedge (\exists p, q : 0 \leq p \leq q \leq N : (\forall i : 0 \leq i < p : v[i] = R)$ 
         $\wedge (\forall j : p \leq j < q : v[j] = B)$ 
         $\wedge (\forall k : q \leq k < N : v[k] = A))$  }

```

Coste: $\Theta(N)$

Algoritmo de partición

Recolocar los elementos de un vector de forma que primero aparezcan los menores que el valor de la primera posición, a continuación los iguales y por último los mayores.

$$\{ N > 0 \wedge v = V \}$$

proc partición(**E/S** $v[0..N]$ **de** ent)

$$\{ v \in Perm(V) \wedge (\exists p, q : 0 \leq p \leq q \leq N : (\forall i : 0 \leq i < p : v[i] < V[0]) \\ \wedge (\forall j : p \leq j < q : v[j] = V[0]) \\ \wedge (\forall k : q \leq k < N : v[k] > V[0])) \}$$

Cambiar en el algoritmo anterior las condiciones por $v[b] < V[0]$, $v[b] = V[0]$ y $v[b] > V[0]$.

Algoritmo de partición

```

{  $N > 0 \wedge v = V$  }
proc partición(E/S  $v[0..N]$  de  $ent$ )
var  $pivote, r, b, a : ent$ 
     $\langle pivote, r, b, a \rangle := \langle v[0], 0, 0, N \rangle ;$ 
    mientras  $(b \neq a)$  hacer
        casos
             $v[b] < pivote \rightarrow \text{intercambiar}(v, b, r) ; \langle r, b \rangle := \langle r + 1, b + 1 \rangle$ 
             $\square v[b] = pivote \rightarrow b := b + 1$ 
             $\square v[b] > pivote \rightarrow \text{intercambiar}(v, b, a - 1) ; a := a - 1$ 
        fcasos
    fmientras
ffun
    {  $v \in Perm(V) \wedge (\exists p, q : 0 \leq p \leq q \leq N : (\forall i : 0 \leq i < p : v[i] < V[0])$ 
         $\wedge (\forall j : p \leq j < q : v[j] = V[0])$ 
         $\wedge (\forall k : q \leq k < N : v[k] > V[0])) \}$ 

```

Coste: $\Theta(N)$