13. (Máxima diferencia)

Dado un arreglo de enteros, calcular la máxima diferencia entre dos de sus elementos (en orden, el primero menos el segundo).

La especificación del programa es:

Const N : Int;

Var a : array[0, N) of Int; r : Int;

{P : N ≥ 2}

S

{Q : r = ⟨Max p, q : 0 ≤ p < q < N : a.p − a.q ⟩}

Paso 1 (invariante)

INV = r = ⟨Max p, q : 0 ≤ p < q < n : a.p − a.q ⟩ ^ 2 ≤ n ≤ N

B = n < N

Luego vale INV ^ ¬B → Q

Paso 2 (inicializamos)

asumimos P

wp.(r,n := E,F).(INV)

≡{ def de wp

E = ⟨Max p, q : 0 ≤ p < q < F : a.p − a.q ⟩ ^ 2 ≤ F ≤ N

={ max no tiene rango vacío, proponemos F = 2, mínima cantidad de elementos

E = ⟨Max p, q : 0 ≤ p < q < 2 : a.p − a.q ⟩

≡{ lógica en el rango

0 ≤ p < q < 2

≡{

0 ≤ p < q ≤ 1

≡{

0 ≤ p ^ p < q ^ q ≤ 1

≡{ por transitividad

(0 ≤ p ^ p < q) → 0 < q ^ q ≤ 1

≡{

(0 ≤ p ^ p < q) → 1 ≤ q ^ q ≤ 1

≡{

(0 ≤ p ^ p < q) → 1 ≤ q ≤ 1

≡{

(0 ≤ p ^ p < q) → q = 1

luego

0 ≤ p ^ p < q ^ q ≤ 1

≡{ transitividad

(p < q ^ q ≤ 1) → 0 ≤ p ^ p < 1

≡{

(p < q ^ q ≤ 1) → 0 ≤ p ^ p ≤ 0

≡{

(p < q ^ q ≤ 1) → p = 0

E = ⟨Max p, q : p = 0 ^ q = 0 : a.p − a.q ⟩

≡{ rango unitario

E = A.0 - A.1

Paso 3 (cota) es lo mismo de siempre weonnnn

Paso 4 (cuerpo del ciclo)

asumimos INV ^ B

luego

wp.(r,n:=E,n+1).(INV)

≡{ def de wp

E = ⟨Max p, q : 0 ≤ p < q < n+1 : a.p − a.q ⟩ ^ 2 ≤ n+1 ≤ N

≡{ logica e hipotesis

E = ⟨Max p, q : 0 ≤ p < q < n +1 : a.p − a.q ⟩

≡{

0 ≤ p < q < n + 1

0 ≤ p < q ^ q < n + 1

0 ≤ p < q ^ q < n v q = n

(0 ≤ p < q ^ q < n) v (0 ≤ p < q ^ q = n)

luego partimos rango, hipótesis por una lado y eliminación de variable por el otro

E = r max ⟨Max p, q : 0 ≤ p < n : a.p − a.n ⟩

≡{ distributiva

E = r max (⟨Max p, q : 0 ≤ p < n : a.p⟩ - a.n)

NO SE PUEDE SEGUIR AAAA, fortalecemos

INV’ = INV ^ r2 = ⟨Max p, q : 0 ≤ p < n : a.p⟩

inicializamos (nuevamente)

para r y n es igual que antes. Pero con n = 2 en el rango de r2 tenemos que 0 ≤ p < 2

es igual a 0 ≤ p ≤ 1, luego p solo puede ser 0 o 1, aplicamos termino y nos queda

r2 = A.0 máx A.1

Cuerpo del ciclo (nuevamente)

para r y n es el mismo proceso pero para r2 tenemos (asumiendo INV’)

F = ⟨Max p, q : 0 ≤ p < n+1 : a.p⟩

≡{

F = ⟨Max p, q : 0 ≤ p < n v p = n : a.p⟩

≡{ partición de rango

F = ⟨Max p, q : 0 ≤ p < n : a.p⟩ max ⟨Max p, q : p = n : a.p⟩

≡{ hipótesis, rango unitario

F = r2 max A.n

**Finalmente**

Const N : Int;

Var a : array[0, N) of Int;

Var r,r2,n : Int;

{P : N ≥ 2}

r,r2,n := A.0 - A.1, A.0 max A.1, 0

do (n < N) →

r,r2,n := r max (r2 - A.n), r2 max A.n, n + 1

od

{Q : r = ⟨Max p, q : 0 ≤ p < q < N : a.p − a.q ⟩}

0 ≤ p < q < 2

≡{ lógica

0 ≤ p < q ≤ 1

≡{ lógica

0 ≤ p ^ p < q ^ q ≤ 1

≡{ asociamos

(0 ≤ p ^ p < q) ^ q ≤ 1

≡{ por transitividad

(0 ≤ p ^ p < q) → 0 < q ^ q ≤ 1

≡{ lógica

(0 ≤ p ^ p < q) → 1 ≤ q ^ q ≤ 1

≡{ lógica

(0 ≤ p ^ p < q) → 1 ≤ q ≤ 1

≡{ lógica

(0 ≤ p ^ p < q) → q = 1

luego lo mismo pero asociando (p < q ^ q ≤ 1) ^ 0 ≤ p

0 ≤ p ^ p < q ^ q ≤ 1

≡{ transitividad

(p < q ^ q ≤ 1) → 0 ≤ p ^ p < 1

≡{

(p < q ^ q ≤ 1) → 0 ≤ p ^ p ≤ 0

≡{

(p < q ^ q ≤ 1) → p = 0