

Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática, de Computadores y del Software (grupo A)

Examen Parcial, 10 de Febrero de 2014.

1. (3 puntos) Especifica, diseña y verifica o especifica y deriva, un algoritmo iterativo de coste lineal que dado un vector v de enteros, un valor entero positivo m tal que $0 \leq m \leq longitud(v)$ y un valor s igual a la suma de las m primeras posiciones del vector, devuelva el valor máximo que se puede obtener sumando m elementos consecutivos del vector.

Por ejemplo, sea $m = 3$, $s = 10$ y el vector

5	-4	9	7	-5	6	-1	10	0	3
---	----	---	---	----	---	----	----	---	---

la suma máxima de 3 elementos consecutivos es $15 = 6 - 1 + 10$.

Justificar el coste de la función implementada.

2. (3 puntos) Un vector de enteros mayores que 0 de longitud 2^n (donde n es un número natural) es *caucásico* si el valor absoluto de la diferencia entre el número de elementos pares de sus mitades es, a lo sumo, 2 y cada mitad también es *caucásica*.

Algunos ejemplos:

- $\{2, 4, 6, 8 \parallel 1, 3, 5, 7\}$ No es *caucásico*, porque su primera mitad tiene 4 elementos pares y la segunda 0.
- $\{2, 4, 6, 8 \parallel 2, 8, 5, 10\}$ Es *caucásico*.
- $\{2, 4, 8, 12, 3, 7, 9, 21 \parallel 10, 20, 30, 1, 3, 5, 7, 40\}$ No es *caucásico* ya que la primera mitad no lo es.

Diseña un algoritmo *recursivo* que determine si un vector de longitud 2^n es *caucásico*. El algoritmo debe de ser eficiente. Determina justificadamente la complejidad.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

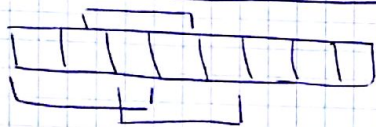
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Febrero 2014

① v vector de enteros
 $0 \leq m \leq \text{long}(v)$

$$s = \sum_{k=0}^{m-1} v[k]$$

Valor máximo que se puede obtener sumando m enteros consecutivos.



Especificación

$\{ 0 \leq n \wedge n = \text{long}(v) \wedge 0 \leq m \leq n \wedge s = \sum_{k=0}^{m-1} v[k] \}$

fn $vMax$ (int $v[]$, int n , int m , int s)

return int $sMax$;

$\{ sMax = \max w : 0 \leq w \leq n-m \cdot (\sum_{t: 0 \leq t \leq m} v[t+w]) \}$

w	t	$v[t+w]$
0	0	$v[0]$
	1	$v[1]$
	2	$v[2]$
1	0	$v[1]$
	1	$v[2]$
	2	$v[3]$

w	t	$v[w+t]$
$n-3$	0	$v[n-3]$
	1	$v[n-2]$
	2	$v[n-1]$

Diseño

int $vMax$ (int $v[]$, int n , int m , int s) {

int $sMax$, i ;
 $sMax = s$;

return $sMax$;

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

Verificación

Invariante $I \equiv (0 \leq i \leq n-m) \wedge$

$$(s = \sum_{k=i}^{i+m-1} v[k]) \wedge$$

$$(s_{max} = \max_{0 \leq l \leq i} \sum_{k=l}^{l+m-1} v[k])$$

(I.1) $\{P\} s_{max} = s; i=0 \{I\}$

pmc $(s_{max} = s, i=0; I) \Leftrightarrow (I)_i^s$

$$\Leftrightarrow (0 \leq 0 \leq n-m) \wedge (s = \sum_{k=0}^{m-1} v[k]) \wedge (s = \max_{0 \leq l \leq 0} \sum_{k=l}^{l+m-1} v[k])$$

$$\Leftrightarrow (0 \leq n-m) \wedge (s = \sum_{k=0}^{m-1} v[k])$$

$$P \equiv (0 \leq n) \wedge (n = \text{length } v) \wedge (s = \sum_{k=0}^{m-1} v[k]) \wedge (0 \leq m \leq n)$$

\Rightarrow pmc \checkmark

(I.2) $\{I \wedge i < n-m\} A_1 \{I\}$

caso del bucle (IF!)

(a) $\{I \wedge (i < n-m) \wedge (s_{max} < s)\} s_{max} = s; s = s - v[i] + v[i+m]; i++; \{I\}$

(b) $\{I \wedge (i < n-m) \wedge (s_{max} > s)\} s = s - v[i] + v[i+m]; i++; \{I\}$

(a) pmc $(\dots, I) \Leftrightarrow (I)_i^{i+1} \left| \begin{array}{l} s - v[i] + v[i+m] \\ s \end{array} \right| \begin{array}{l} s \\ s_{max} \end{array}$

$$\Leftrightarrow (0 \leq i+1 \leq n-m) \wedge (s - v[i] + v[i+m] = \sum_{k=i+1}^{i+m} v[k]) \wedge$$

$$\wedge (s = \max_{0 \leq l \leq i+1} \sum_{k=l}^{l+m-1} v[k])$$

$$\Leftrightarrow (0 \leq i+1 \leq n-m) \wedge (s = \sum_{k=i+1}^{i+m-1} v[k]) \wedge (s = \max_{0 \leq l \leq i+1} \sum_{k=l}^{l+m-1} v[k])$$

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

$\wedge (i < n-m) \wedge (s_{max} < s)$

(I3)

$$I \wedge (i > n-m) \Rightarrow (s_{\max} = \max_{0 \leq l \leq n-m} \sum_{k=l}^{l+m-1} v[k])$$

$$(P > n-m) \wedge (0 \leq i \leq n-m) \wedge (s = \sum_{k=i}^{i+m-1} v[k]) \wedge (s_{\max} = \max_{k=i}^{i+m-1} v[k])$$

$$\Rightarrow (i = n-m) \wedge (s_{\max} = \max_{0 \leq l \leq i} \sum \dots)$$

$$\Rightarrow s_{\max} = \max_{0 \leq l \leq n-m} \sum_{k=l}^{l+m-1} v[k]$$

Función de cota

$$C(v, n, m, s, i, s_{\max}) = n - m - i$$

$$(C1) I \wedge (i < n-m) \Rightarrow n-m-i > 0 \Rightarrow n-m-i > 0 \checkmark$$

$$(C2) \{ (I \wedge B) \wedge T = c \mid A \} \{ t > c \}$$

Podemos hacerlo de una vez por en el if no tocamos i.

$$\text{pod } (A_s, t > c) \Leftrightarrow (t > n-m-1) \left| \begin{array}{l} s \\ s_{\max} \end{array} \right| \left. \begin{array}{l} s-v[i]+v[i+m] \\ i \end{array} \right|^{l+i}$$

$$(A) t > n-m - (i+1) \Leftrightarrow t > n-m-i-1$$

COMPLEJIDAD

$$T(n) = C(n-m-1) + \bar{c}$$

$$= Cn + cte = O(n)$$



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

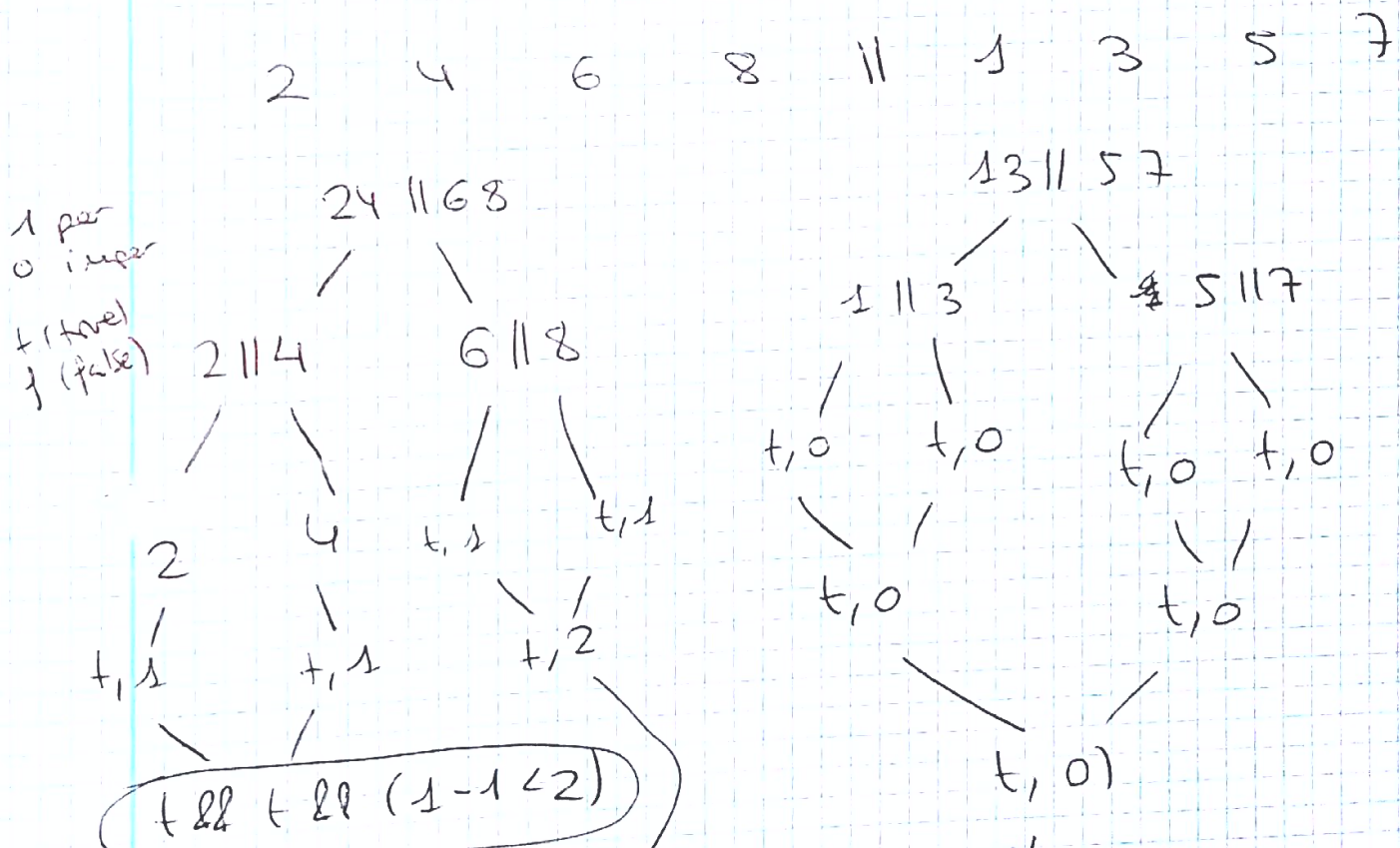
② $2468 \parallel 1357 \Rightarrow$ no es canónico
 4 pares 0 pares

$2|4|68 \parallel 28|510 \Rightarrow$ sí es canónico
 4 3

Planteamiento

• Caso base : un elemento \Rightarrow es canónico.

• Caso recursivo



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



```
void esCausésico (int v [ ], int P, int U, bool &causa,
int & nPares)
```

```
bool causa I, causa D;
int nPares I, nPares D, M;
```

```
if (P == U) {
```

```
causa = true;
```

```
if (v [ P ] % 2 == 0) nPares = 1;
```

```
else nPares = 0;
```

```
} else {
```

```
M = (P + U) / 2;
```

```
esCausésico (v, P, M, causa I, nPares I)
```

```
esCausésico (v, M + 1, U, causa D, nPares D)
```

```
causa = causa I && causa D && (abs(nPares I - nPares D) <= 2)
```

Complejidad

Recursión por división
(caso base)

$$T(n) = a T(n/b) + cn^k$$

en nuestro caso $a=2, b=2, k=0,$

$$O(n^{\log_b a}) = a > b^k$$

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

3

total artistas (Vuelta Atmos)
 $M > N$ ← n: escenarios

vetos $[M] [M]$

beneficios

(cada artista tiene un beneficio)

Planteamiento

La solución viene dada por:
un vector de M posiciones una por cada artista de booleanos que indican si los artistas participan en el festival o no.

• Restricciones explícitas:

Lista de M booleanos

• Restricciones implícitas:

- Maximizar el beneficio
- Respetar los vetos
- Asisten exactamente n artistas así que hay n trues y $M-n$ falses

PARÁMETROS

- bool sol $[M]$: vector solución actual
- int N : Num artistas en el festival
- int M : Num artistas totales.
- bool veto $[M] [M]$: matriz de vetos
- solOpt $[M]$: vector sol óptimo

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

Código:

```
void festival (bool sol [], bool solOpt [], int &bAct,  
int &bMax, int k, int &nAct) {
```

```
sol[k] = true; // Metemos en el festival  
          el artista
```

```
bAct += benef[k];
```

```
nAct ++;
```

```
if (esValida(sol, k, nAct) // que no haya vetos  
                          // Es sol
```

```
    if (k == M - 1 || (N == nAct)  
                        // n artistas vienen
```

```
        if (bAct > bMax) // Beneficios mejores?
```

```
            bMax = bAct;
```

```
            CopiaSol(solOpt, sol);
```

```
    else { // Si no es sol
```

```
        if (k < M - 1) // Elijo el siguiente si se puede elegir  
            festival(sol, sol, bAct, bMax, k + 1, nAct);
```

```
sol[k] = false; // Si no, quito al artista
```

```
bAct -= benef[k];
```

```
nAct --;
```

```
if (k == M - 1 || (nAct == N)) // Lo volveremos  
                                a comprobar
```

```
    if (bAct > bMax)
```

```
        bMax = bAct
```

The logo for Cartagena99, featuring the text "Cartagena99" in a stylized, green and blue font with a shadow effect.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70


```
bool esValida (int sol [], int k, int nAct)
```

```
bool v = false;
```

```
for (int i = 0; i < k; i++)
```

```
    v = v || (veto [i] [k])  
           || sol [i];
```

} Una vez se
han encendido
veto → true

```
return (v || nAct <= N)
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70