

Ejercicios Tema 2 DyV

Julio 2022

Ejercicio 4

4. (2 puntos) Sea C un conjunto de n elementos desordenado. Deseamos construir un algoritmo **Divide y Vencerás** que, sin necesidad de ordenar el conjunto, nos diga qué elemento del conjunto estaría exactamente en la posición $\lfloor n/2 \rfloor$ si el conjunto estuviese ordenado (nota: $\lfloor x \rfloor$ indica parte entera de x). Por ejemplo si $n = 9$ (y $\lfloor 9/2 \rfloor = 4$) y el conjunto $(4, 7, 6, 1, 2, 10, 8, 11, 3)$ lo representamos en un array $(v[0], v[1], \dots, v[8])$, el elemento que estaría en la posición central $v[4]$ sería 6.



algoritmo-DyVC $\text{int}^*v, \text{int } ini, \text{int } fin > h$

$\text{int } n = fin - ini + 1$

if $C n < 5 > h$ bubblesort($C v$)

return $v[ini + n/2]$

else h

$\text{int } m1 = \text{algoritmo-DyVC } v, ini, ini + n/3 >$

$\text{int } m2 = \text{algoritmo-DyVC } v, ini + n/3, ini + 2 * n/3 >$

$\text{int } m3 = \text{algoritmo-DyVC } v, ini + 2 * n/3, fin >$

if $C m1 >= m2 >$

return $C m1 <= m3 > ? m1 : C m3 >= m2 > ? m3 : m2$

else

return $C m2 <= m3 > ? m2 : C m3 >= m1 > ? m3 : m1$

h

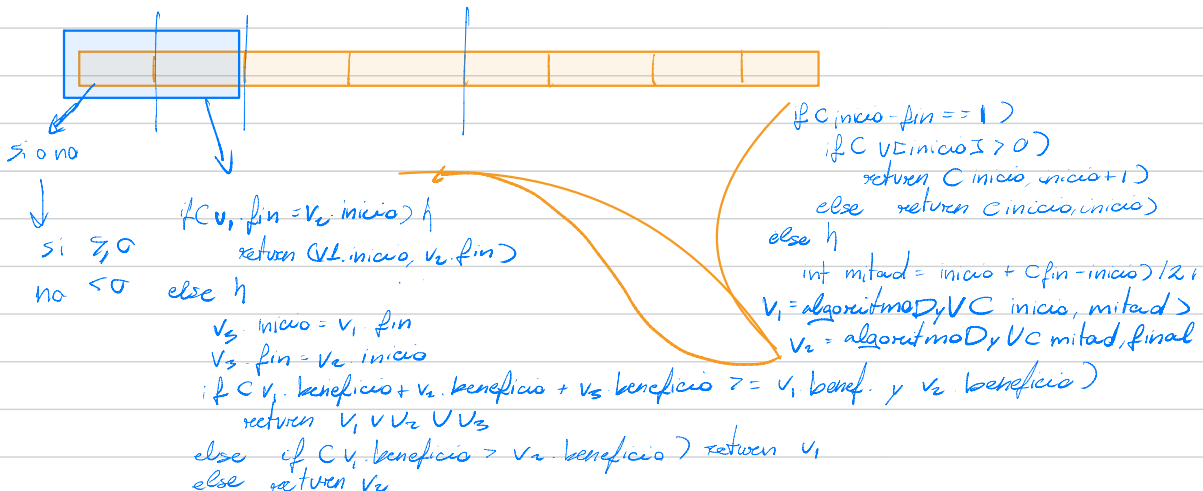
h

5. (2 puntos) Un aficionado al póker juega una partida cada día, y anota los euros que gana o pierde. Por ejemplo, la siguiente tabla muestra los resultados del último mes.

L	M	X	J	V	S	D
					29	-7
14	21	30	-47	1	7	-39
23	-20	-36	-41	27	-34	7
48	35	-46	-16	32	18	5
-33	27	28	-22	1	-20	-42

Podemos ver que empezó el mes con una ganancia de 29 euros, y terminó con una pérdida de 42. El beneficio total obtenido en el mes (la suma de todas las ganancias y pérdidas) es -50 euros. Analizando la información, el jugador se podría dar cuenta de que si hubiera empezado a jugar el día 16 y terminado el día 26, habría maximizado sus ganancias, obteniendo 105 euros.

En general, dado un vector de ganancias/pérdidas de longitud n , se desea encontrar el subvector sobre el cual se consigue el beneficio total máximo. Diseñad un algoritmo que realice esta tarea de forma lo más eficiente posible.



3. (2 puntos) Diseña un algoritmo para calcular x^n , $n \in \mathbb{N}$, con un coste $O(\log n)$ en términos del número de multiplicaciones.

```
algoritmo  $D_V C x, n$  h
  if  $C n == 1$  return  $x$ 
  else h
    pot_mitad =  $D_V C x, n/2$ 
    if  $C n \% 2 == 0$ 
      return  $pot\_mitad * pot\_mitad$ 
    else
      return  $x * pot\_mitad * pot\_mitad$ 
  f
f
```