

ALGORITMOS DEVORADORES



8 DE MARZO DE 2018

GEMA RICO POZAS

UO238096

TRABAJO PEDIDO 1

a) Razonar qué ladrón propone mejor idea (heurístico).

El ladrón que propone la mejor forma de llevar a cabo el robo es el ladrón número 3.

b) ¿Cree que esa idea hace el robo óptimo para cualquier surtido de joyas (n , p , v y k)?

Sí, puesto que tenemos que introducir en la mochila de forma que la suma de los beneficios de los elementos escogidos sea máxima.

c) Implementar el algoritmo seleccionado anteriormente de tres formas:

1. Implementación₁: Ir seleccionando las joyas según lo fijado, pero sin hacer ninguna ordenación previamente.
2. Implementación₂: Hacer ordenación (pero mediante un algoritmo de los "malos" $O(n^2)$), para después ir seleccionando según lo fijado.
3. Implementación₃: Ordenar con un algoritmo eficiente ($O(n \log n)$), para después seleccionar las joyas una vez ordenadas según lo fijado. Así sí que acompañaremos el buen heurístico con una buena implementación.

Ver en el proyecto adjunto.

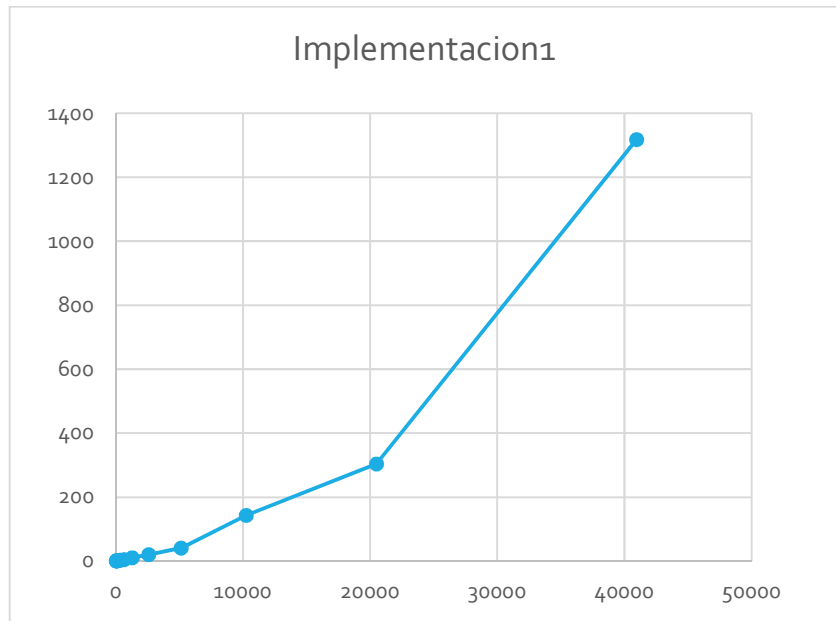
d) Calcule la complejidad de cada una de las tres implementaciones anteriores.

1. Implementación₁: $O(n^2)$
2. Implementación₂: $O(n^2)$
3. Implementación₃: $O(n \log n)$

e) Haga una medición de tiempos de ejecución de cada una de las tres implementaciones anteriores. Se debe ir creciendo el tamaño n así: 10, 20, 40, 80, ..., hasta que se desborde la memoria HEAP. Para cada tamaño n , la generación tanto de cada elemento p_i , como de cada v_i , serán valores enteros aleatorios en el rango [10...99] y el valor de k será en cada caso $k=25*n$.

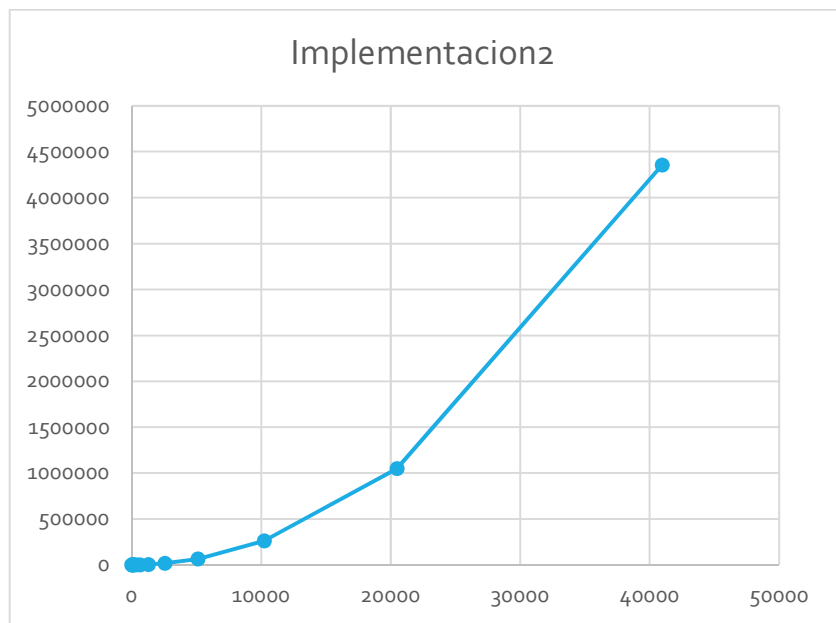
Implementación 1

Carga de trabajo (n)	tiempo en micros
10	1
20	1
40	1
80	2
160	2
320	3
640	5
1280	11
2560	21
5120	41
10240	143
20480	304
40960	1318



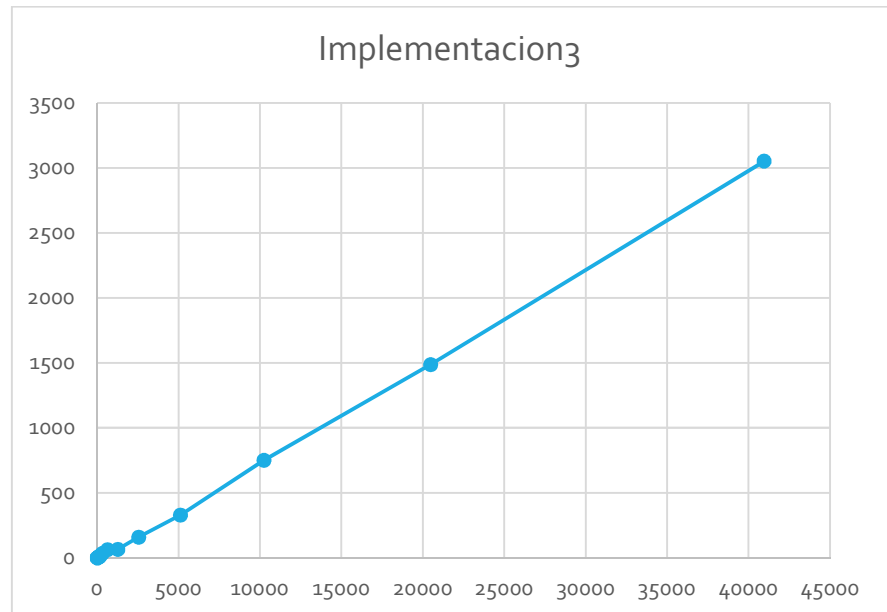
Implementación 2

Carga de trabajo (n)	tiempo en micros
10	2
20	3
40	8
80	17
160	64
320	255
640	1019
1280	4065
2560	16274
5120	65271
10240	260971
20480	1047564
40960	4357137



Implementación 3

Carga de trabajo (n)	tiempo en micros
10	2
20	2
40	2
80	9
160	12
320	35
640	64
1280	66
2560	159
5120	331
10240	751
20480	1487
40960	3052



f) Compruebe si los tiempos obtenidos en el apartado anterior concuerdan o no con la complejidad teórica, en cada uno de los tres casos o implementaciones.

Como podemos apreciar en las gráficas las complejidades de los algoritmos coinciden con lo establecido en el punto d).