

Ejercicios Unidad 1 – Prog. Avanzada

6.

A continuación se va a analizar el coste del algoritmo.

ALGORITMO bucketSort(V:vector[1..n] de Real)

Aux: bucket: vector[0..n- 1] de listas //Coste contante (1)

Metodo:

Para i ← 1 hasta n hacer //Coste n

c ← calculaBucket (V[i]) //Coste 1

inserta(bucket[c], V[i]) //Coste 1

fpara

Para i ← 0 hasta n-1 hacer //Coste n

ordenar(bucket[i]) //Coste 1 gracias a que hay

distribución normal

fpara

V ← concatenar(bucket[0], ... , bucket[n-1]) //Coste n

Coste: $1 + n + n + n = 3n + 1$

$O(n)$

Sin distribución normal: $1 + n + n * n^2 + n = n^3 + n + 1$

$O(n^3)$

7. a)

El coste del algoritmo en el peor de los casos sería llamar al algoritmo n veces y dentro del bucle se realizan como mucho m iteraciones. $O(n*m)$. Como $n = 2^m - 1$ podemos hacer también $m = \log_2(n+1)$.

El coste para el peor de los casos es de $O(n*\log_2(n+1))$