## 5.4

- a) La logarítmica
- b) La cuadrática
- c)La cuadrática
- d)No, va depende de la entrada dada para entradas de 5 ya empieza a ser más optima la logarítmica

## 5.5

- a) No llega a retornar el índice de la clave final, sale antes de entregarlo
- b) En algún momento puede intentar recorrer un índice que no existe
- c)Se hacen más ciclos en el caso de que el elemento se encuentre después del medio actual.
- d) Se hacen más ciclos en el caso de que el elemento se encuentre antes del medio actual.

## 5.6

- a) O(n) Ya que sumo cada cifra (puede haber acarreo, pero esto no influye demasiado)
- b) O(n\*\*2) Multiplica cifra por cifra sería como hacer un bucle anidado dentro de otro
- c)O(n\*\*2) Ocurre lo mismo que con la multiplicación solo que es el inverso de este

#### 5.10

- a)2,5
- b)3,37ms
- c)12,5ms
- d)62,5ms

## 5.11

a)12.000.000 de tamaño de entrada

- b)3.500.000 de tamaño de entrada
- c)34.641 de tamaño de entrada

d)4.932

## 5.12

N	O(N**3)	o(N**2)	ON(log N)	O(N)
10	0,00103	0,00045	0,00066	0,0034
100	0,47015	0,01112	0,00486	0,00063
1000	448,77	1,1233	0,05843	0,00333
10000	Aprox.448770	111,13	0,68631	0,03042
100000	Aprox.4,49x10^8	Aprox.11113	8,0113	0,29832
1000000	Aprox.4,49x10^11	Aprox. 1111300	95,9	2,98

No hay cambios repentinos en el crecimiento de todos los órdenes de ejecución.

El más optimo siempre va a ser O(N) si es posible y O(NlogN) no es una mala alternativa

N\*\*2 y N\*\*3 para números grandes son poco prácticos

## 5.13

- 1. 2/N si el valor de n es lo suficientemente grande ya es más pequeña que la contantes siguiente
- 2. 37
- 3. Raíz de N
- 4. N
- 5. N log log N
- 6. N  $\log N$  en crecimiento son igual por propiedades del logaritmo N  $\log N^{**}2$ . Ya que el 2 queda por fuera del logaritmo siendo una constante
- 7. N log\*\*2 N
- 8. N\*\*2
- 9. N\*\*2 log N
- 10. N\*\*3
- 11.2\*\*(N/2)
- 12.2\*\*N

# 5.16

- a) La multa sería de 2\*\*(2\*\*(N-1))
- b)Los días necesarios para esto serían log log E(euros)