

## Exercícios de Revisão

1. Faça uma análise do tempo de execução (notação Big-O) para cada um dos 4 programas a seguir. Observe que o tempo de execução corresponde aqui ao número de vezes que a operação `sum++` é executado. `sqrt` é a função que retorna a raiz quadrada de um determinado número.

- (a) `sum = 0;`  
`for(i=0; i<sqrt(n)/2; i++)`  
    `sum++;`  
`for(j=0 ; j<sqrt(n)/4; j++)`  
    `sum++;`  
`for(k=0; k<8+j; k++)`  
    `sum++;`
- (b) `sum = 0;`  
`for(i=0; i<sqrt(n)/2; i++)`  
    `for(j=i; j<8+i; j++)`  
        `for(k=j; k<8+j; k++)`  
            `sum++;`
- (c) `sum = 0;`  
`for(i=1; i<2*n; i++)`  
    `for(j=1; j<i*i; j++)`  
        `for(k=1; k<j; k++)`  
            `if (j % i == 1)`  
                `sum++;`
- (d) `sum = 0;`  
`for(i=1; i<2*n; i++)`  
    `for(j=1; j<i*i; j++)`  
        `for(k=1; k<j; k++)`  
            `if (j % i)`  
                `sum++;`

2. Compare a ordem assintótica de crescimento dos seguintes pares de funções. Em cada caso diga se:

$f(n) \in \Theta(g(n))$ ,  $f(n) \in O(g(n))$  or  $f(n) \in \Omega(g(n))$ .

	$f(n)$	$g(n)$
a)	$100n + \log n$	$n + (\log n)^2$
b)	$\log n$	$\log n^2$
c)	$\frac{n^2}{\log n}$	$n(\log n)^2$
d)	$(\log n)^{\log n}$	$\frac{n}{\log n}$
e)	$\sqrt{n}$	$(\log n)^5$
f)	$n2^n$	$3^n$
g)	$2^{\sqrt{\log n}}$	$\sqrt{n}$

3. O vetor com os seguintes valores 23,17,14,6,13,20,10,11,5,7,12 é um heap? Se não, converta-o usando o algoritmo de construção de heap visto em aula.