Universidade Federal de Ouro Preto
Departamento de Computação - DECOM
BCC241 – Projeto e Análise de Algoritmos
Prof. Anderson Almeida Ferreira
Exercícios – Teste de Primalidade

Turma: 11
Nome: Pedro Lucas Damasceno Silva
Matrícula: 20.1.4003

1.

divide(44,5)	(8,4)
divide(22,5)	(4,2)
divide(11,5)	(2,1)
divide(5,5)	(1,0)
divide(2,5)	(0,2)
divide(1,5)	(0,1)
divide(0,5)	(0,0)

2. if (y = 0) return 1; **O(n)**
   z = modexp(x, floor(y/2), N); **Pior caso na divisão por 2: n shifts**
   if (y % 2 = 0) **$O(n^2)$**
   return z^2 mod N; $n^2 + n^2$ = **$O(n^2)$**
   else
   return x . z^2 mod N; $n^2 + n^2$ = **$O(n^2)$**

   No pior caso, são feitos n shifts à direita. Multiplicando pelo custo local (n^2), temos: n . $O(n^2)$ = **$O(n^3)$**.

3. Pick positive integers a1, a2, ..., ak < N at random; **k**
   if ($a_i^{N-1} \equiv 1$ (*mod* N)) for all i = 1, 2, ..., k; **$O(n^3)$***
   return yes; **O(1)**
   else
   return no; **O(1)**

   *No pior caso, todo laço de repetição é executado, realizando k + 1 comparações. Multiplicando pelo custo de *mod*, temos: $kO(n^3) + n(k+1)$, considerando a comparação de custo n. Eliminando as constantes e prevalecendo o termo de maior grau: **$O(n^3)$**.