**( a )**

**int letraA(int n, int a) {**

**int i; 1**

**float resultado = 0, aux; 1**

**for (i = 1; i <= n; i++) n + 1**

**{**

**aux = i / pow(a, i); n**

**resultado = resultado + aux; n**

**}**

**return resultado; 1**

**}**

**T(n) = 1 + 1 + (n + 1) + n + n + 1**

**T(n) = (n+1) + (n+1) + (n + 1) + 1**

**T(n) = 3(n+1) + 1 ou T(n) = 3n +4**

**=> O(n)**

**( b )**

**int letraB(int n, int m) {**

**int i, j; 1**

**int resultado = 0; 1**

**for(i = 1; i <= n; i++) n + 1**

**{**

**for(j = 1; j <= m; j++) n . (m + 1)**

**{**

**resultado = resultado + (i \* j); n . m**

**}**

**}**

**return resultado; 1**

**}**

**T(n,m) = 1 + 1 + (n+1) + [n.(m+1)] + (n.m) + 1**

**T(n,m) = [n.m + n] + (n.m) + n + 4**

**T(n,m) = 2n.m + 2n + 4**

**=> O(n.m)**

**( c )**

**int letraC(int n) {**

**int i; 1**

**for(i = 2; i < n; i++) n – 1**

**{**

**if(n == 2 || n == 1) n – 1**

**return 1; 0**

**if(n % i != 0) n – 1**

**continue; n – 2**

**else n – 1**

**return 0; 0**

**}**

**return 1; 1**

**}**

Os Piores casos são os números primos, onde “if(n % i != 0)” permite que entrem n-2 vezes

|  |  |  |
| --- | --- | --- |
| N | **#vezes** |  |
| 1 | **Não chega** |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |

**T(n) = 1 + (n-1) + (n-1) + (n-1) + (n-2) + (n-2) + (n-1) + 1**

**T(n) = 4(n-1) + (n-2) + 2**

**T(n) = 4n – 4 + n – 2 + 2**

**T(n) = 5n – 4**

**=> O(n)**