

$$T(0) = O(1) \quad \forall c = 1, n_0 = 1.$$

Problem 21:  $T(n) = \begin{cases} 3T(n-1), & \text{if } n > 0 \\ 1, & \text{otherwise.} \end{cases}$

Solution:

$$T(n) = 3T(n-1)$$

$$T(n) = 3(3T(n-2)) = 3^2 T(n-2)$$

$$T(n) = 3^2 (3T(n-3)) = 3^3 T(n-3)$$

...

$$T(n) = 3^n T(n-n) = 3^n T(0) = 3^n$$

$$\Rightarrow O(3^n).$$

Problem 22:  $T(n) = \begin{cases} 2T(n-1) - 1, & \text{if } n > 0 \\ 1, & \text{otherwise} \end{cases}$

$$T(n) = 2T(n-1) - 1$$

$$T(n) = 2(2T(n-2) - 1) - 1 = 2^2 (T(n-2) - 1)$$

$$T(n) = 2^2 (2(T(n-3) - 1) - 1) = 2^3 (T(n-3) - 2^1 - 2^0)$$

$$T(n) = 2^n - (2^n - 1)$$

$$T(n) = 1$$

$$O(1).$$

Problem 23.

public void function(int n)

{ int i = 1, s = 1;

while (s <= n) {

i++;

s = s + i;

System.out.println(i);

}

$$1 + 2 + 3 + \dots + k = \frac{(k+1)k}{2}$$

$$\frac{k(k+1)}{2} > n$$

$$\Rightarrow k = O(\sqrt{n})$$