Análisis de algoritmos
Quiz #3

Deyner Navarro Badilla
2020113009

```cpp
int busquedaSecuencial(std::vector<int> array, int n, int key) {

    int indexResult = -1;

    for (int i = 0; i < n; i++) {
        if (array[i] == key) {
            indexResult = i;
            break;
        }
    }
    return indexResult;
}
```

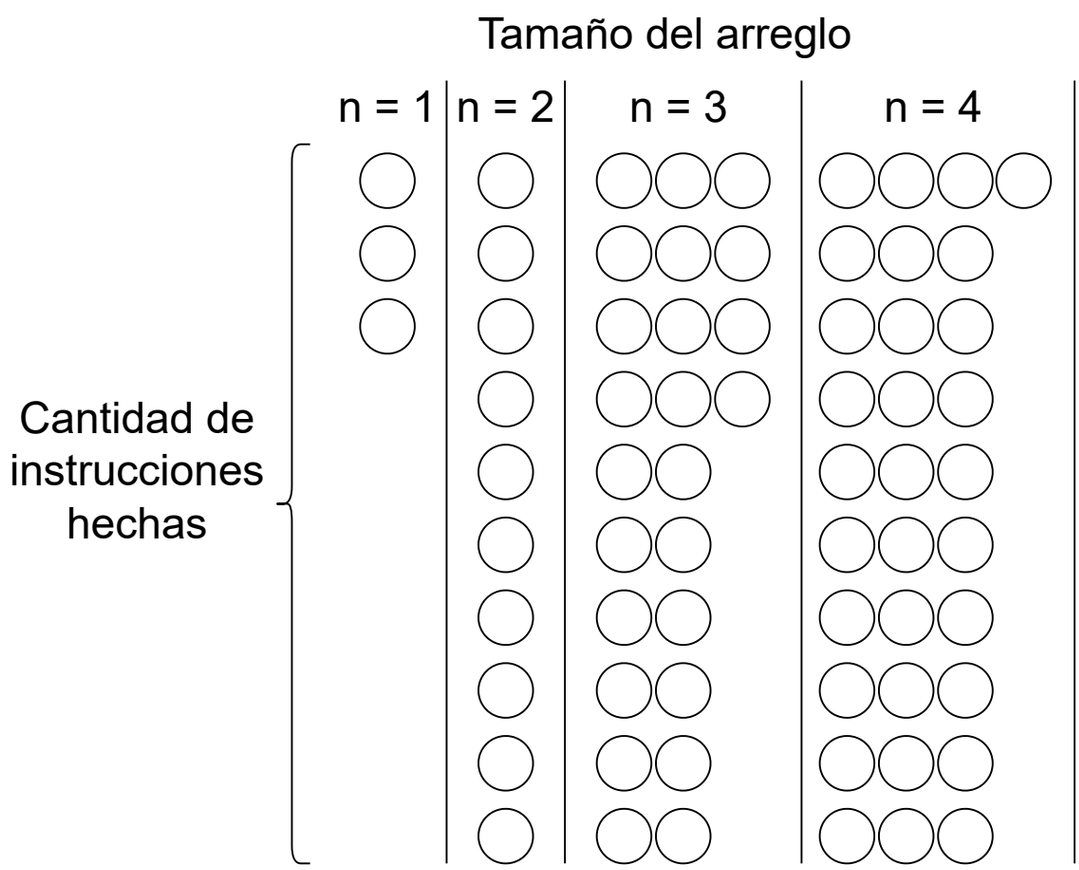
$1+1+(1+2+2+1+1)(n)+1$

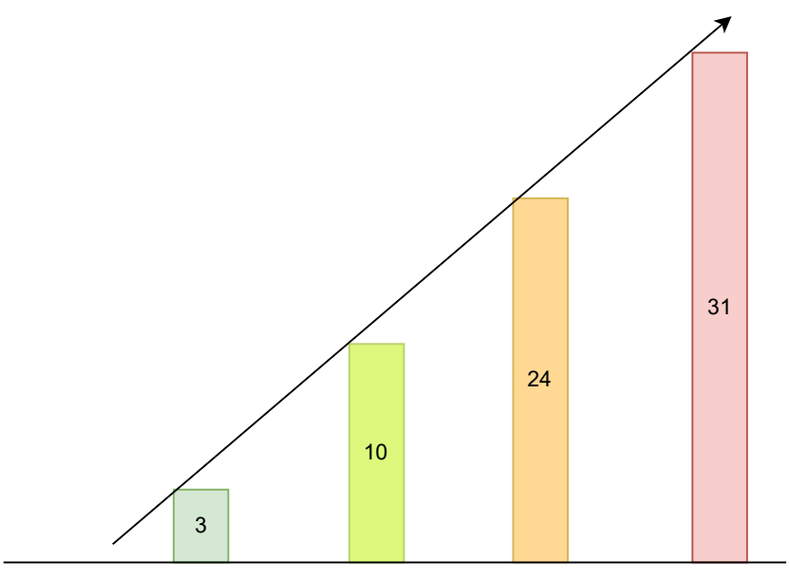
$f(n) = 7n + 3$

$f(n) = O(n)$

$f(n) = O(n)$

El Peor de los casos:





La key esta al final

Para el peor de los casos se puede observar como la cantidad de instrucciones necesarias para resolver el problema va aumentando según el tamaño del arreglo dado.

Asemeja la función

O(g(n)) = O(n)

Lineal

```cpp
int busquedaSecuencial(std::vector<int> array, int n, int key) {
    int indexResult = -1;
    for (int i = 0; i < n; i++) {
        if (array[i] == key) {
            indexResult = i;
            break;
        }
    }
    return indexResult;
}
```

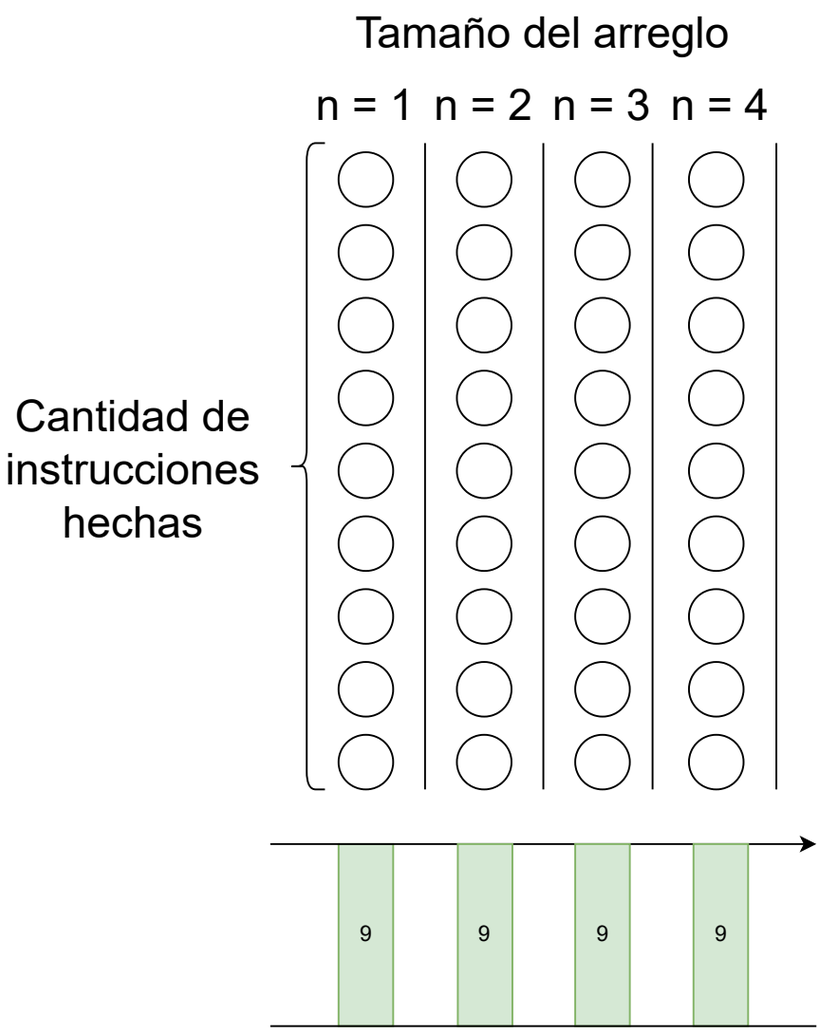
1+1+(1+2+1+1)(n-(n-1))+1

f(n) = 9

f(n) = Ω(c)

f(n) = Ω(c)

El Mejor de los casos:



La key esta al inicio

En el mejor de los casos, será cuando la llave buscada se encuentre al inicio del arreglo. Se puede observar como este mantiene las mismas cantidades de instrucciones necesarias sin importar el tamaño del arreglo.

Asemeja la función

Ω(g(n)) = Ω(c)

Constante