# Análisis de complejidad temporal

|  |  |  |
| --- | --- | --- |
| # | contains() | # veces |
| 1 | if(front == null) | 1 |
| 2 | return false | 1 (solo si entra al if de la línea 1) |
| 3 | QueueNode<T> current = front | 1 |
| 4 | for (int i = 0; i < size; i++) | n |
| 5 | if(current.getValue().equals(elementSearched)) | n-1 |
| 6 | return true | 1 (solo si entra al if de la línea 5) |
| 7 | else if(current.getValue().equals(back.getValue())) | n-1 |
| 8 | return false | 1 (solo si entra al if de la línea 7) |
| 9 | current = current.getNext() | n-2 |
| 10 | return false | 1 |

En caso de que entre al if inicial:

En caso de que no entre al if inicial:

4)

en el 5 se ejecuta la línea 4, pero no entra al for.

, , entra al for.

, , entra al for.

, , entra al for.

, , entra al for.

, , entra al for.

, , no entra al for.

5) La línea se ejecuta solo cuando entra al for. Como el for se ejecuta veces, entra veces. Por lo tanto, esta línea se ejecuta veces.

9) La línea solo se ejecuta cuando entra al for, pero no entra ni al if de la línea 5, ni al de la línea 7. Por lo tanto, habrá un caso en el que entre al for, pero no se ejecute esta línea, pues habrá entrado a alguno de los dos if anteriores. Entonces, la línea se ejecuta veces.

10) La línea solo se ejecuta en caso de que no se entre al for.

En caso de que no entre al for:

Si no entra al for, significa que , y se ejecuta la línea 10.