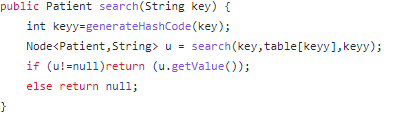
Space complexity



|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Size | Quantity |
| input | Key | 256 bits | 1 |
| aux | keyy | 32 bits | 1 |
| output | u | 1056 bits | 1 |

Complexity =

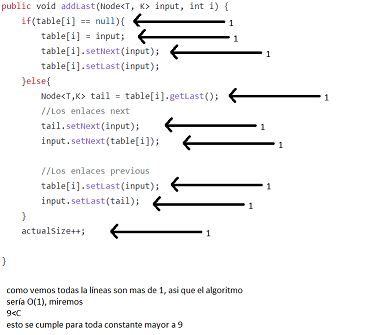
Texto

Descripción generada automáticamente

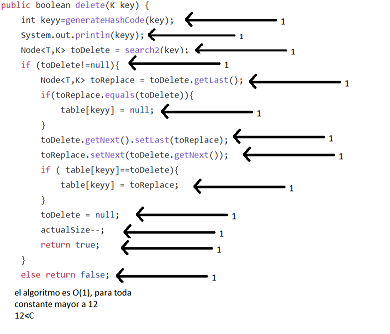
|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Size | Quantity |
| Input | Object | 1056 bits | 1 |
| Input | Head | 1056 bits | 1 |
| aux | n | 1056 bits | 1 |

Complejidad espacial= 3 =

Temporal complexity

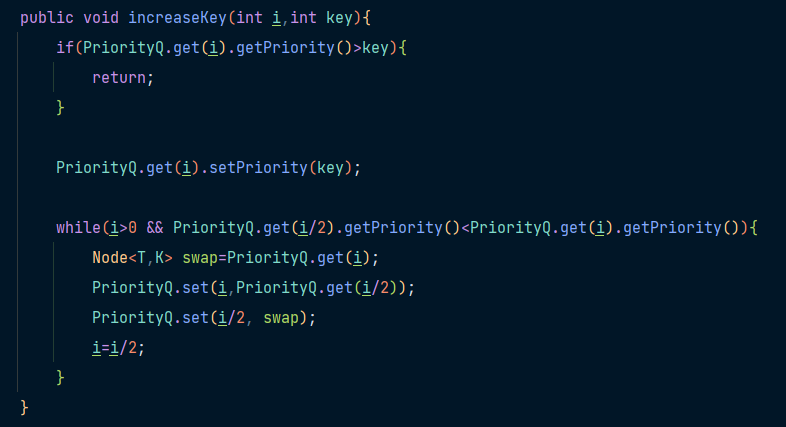


Everything run son constant time, therefore the complexity is O(1)



Everything runs in constant time, therefore the complexity is O(1).

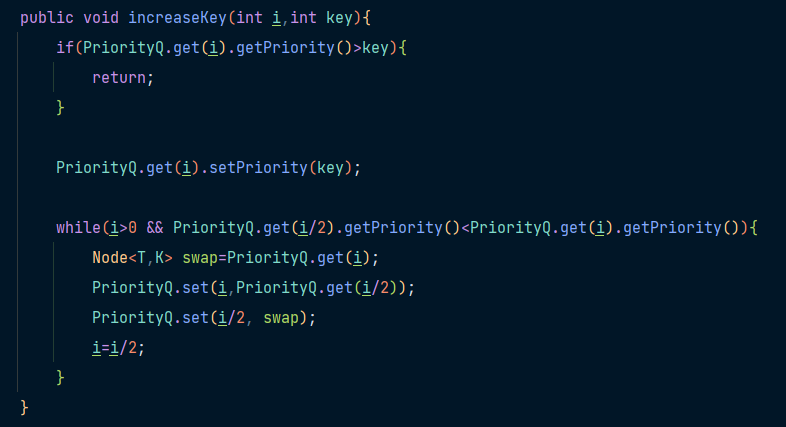
Extra Complexity:



|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Size | Quantity |
| Input | i | 32 bits | 1 |
| Input | key | 32 bits | 1 |
| aux | swap | 1056 bits | 1 |

Complexity =

Time Complexity:



public void increaseKey(int i,int key){

if(PriorityQ.get(i).getPriority()>key){ <-------------------------1

return; <-------------------------1

}

PriorityQ.get(i).setPriority(key); <-------------------------1

while(i>0 && PriorityQ.get(i/2).getPriority()<PriorityQ.get(i).getPriority()){ <-------------

Node<T,K> swap=PriorityQ.get(i); <-------------------------

PriorityQ.set(i,PriorityQ.get(i/2)); <-------------------------

PriorityQ.set(i/2, swap); <-------------------------

i=i/2; <-------------------------

}

}

Now we sum every line value

3 +

Therefore it is O(log(n))