# Agenda

- [15 min] Course evaluation (content only, assessment after)
- Contest for Mock Theory Quiz (when there are >1 correct option, just choose the first one you see)
- Q & A

# Instructions

- Go to https://app.evalytics.nl
- Click on evaluate with code ('evalueren met code')
- Enter this code: ifn-415

# Instructions

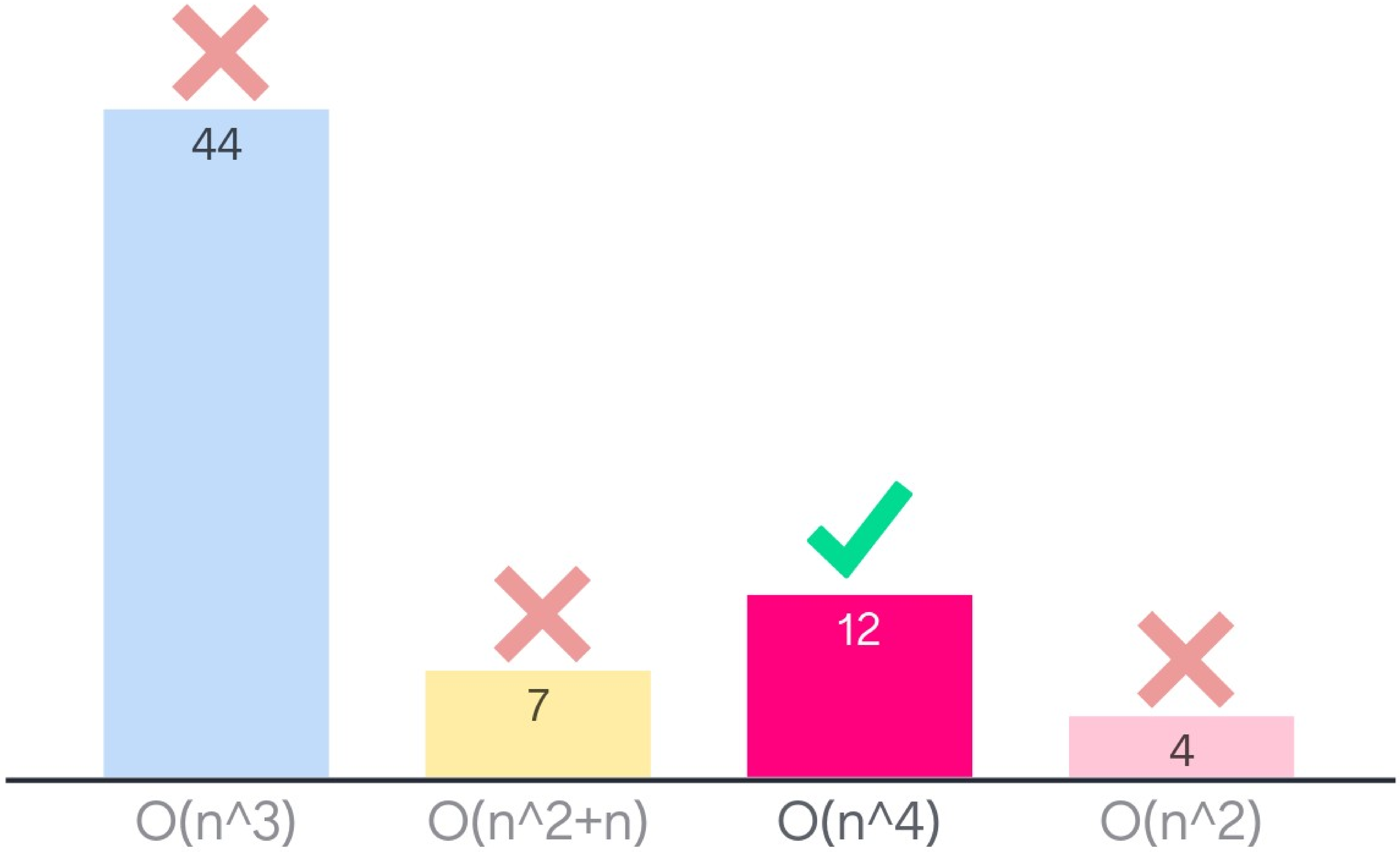
**What is the complexity (expressed with *n*) of the following code snippet (in terms of Big-O notation)?**

```
for(int i=0; i<=n ; i++){
    for(int j=1; j<=i * i; j++){
        if (j % i == 0){
            for(int k = 0; k<j; k++){
                sum++;
            }
        }
    }
}
```

- [ ] O(n^4)
- [ ] O(n^2)
- [ ] O(n^3)
- [ ] O(n^2+n)

26

# What's the complexity?

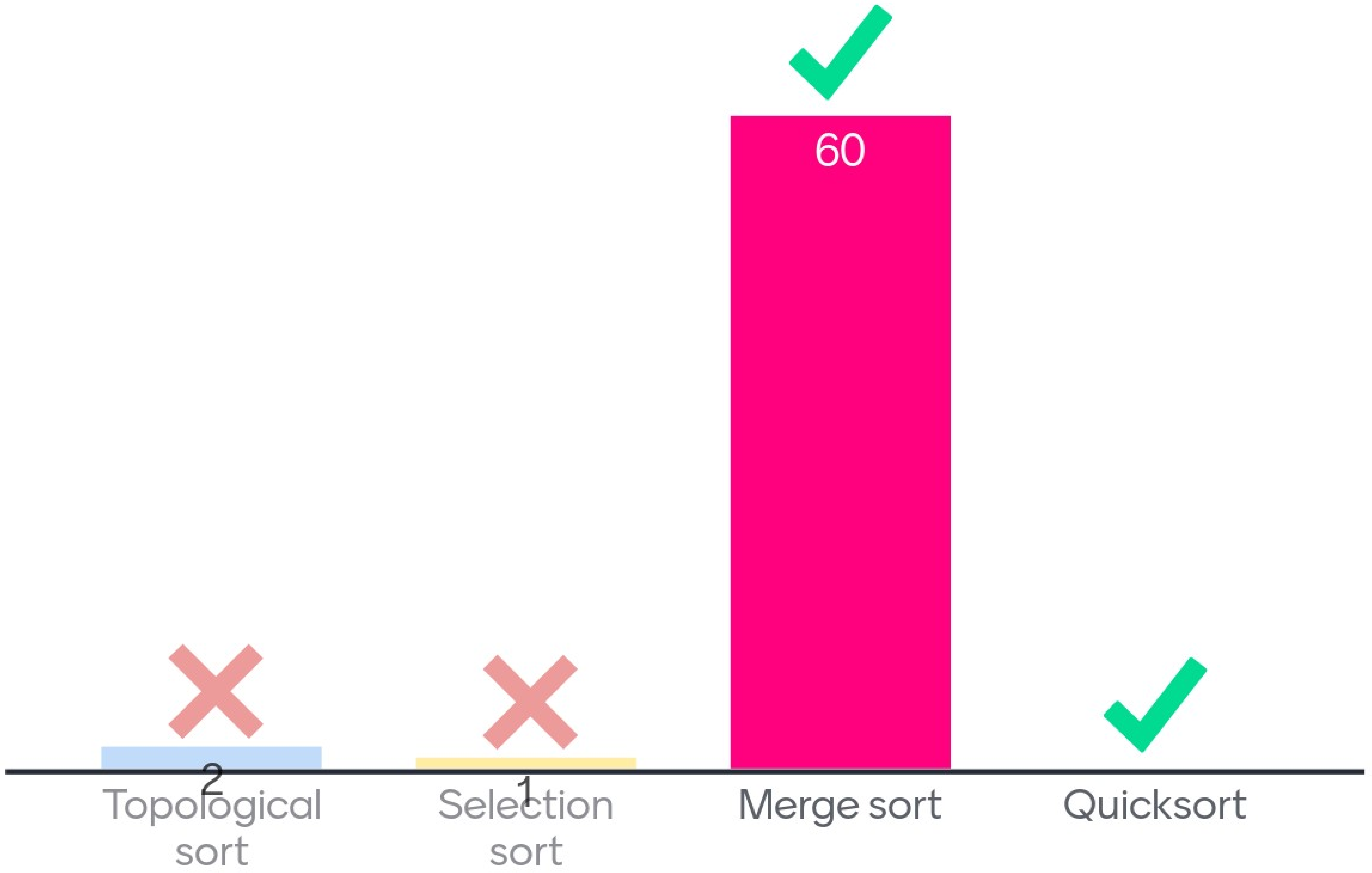


**Which of the following algorithms (seen in class) follow the divide-and-conquer paradigm?**

- [ ] Merge sort
- [ ] Quick sort
- [ ] Selection sort
- [ ] Topological sort

3

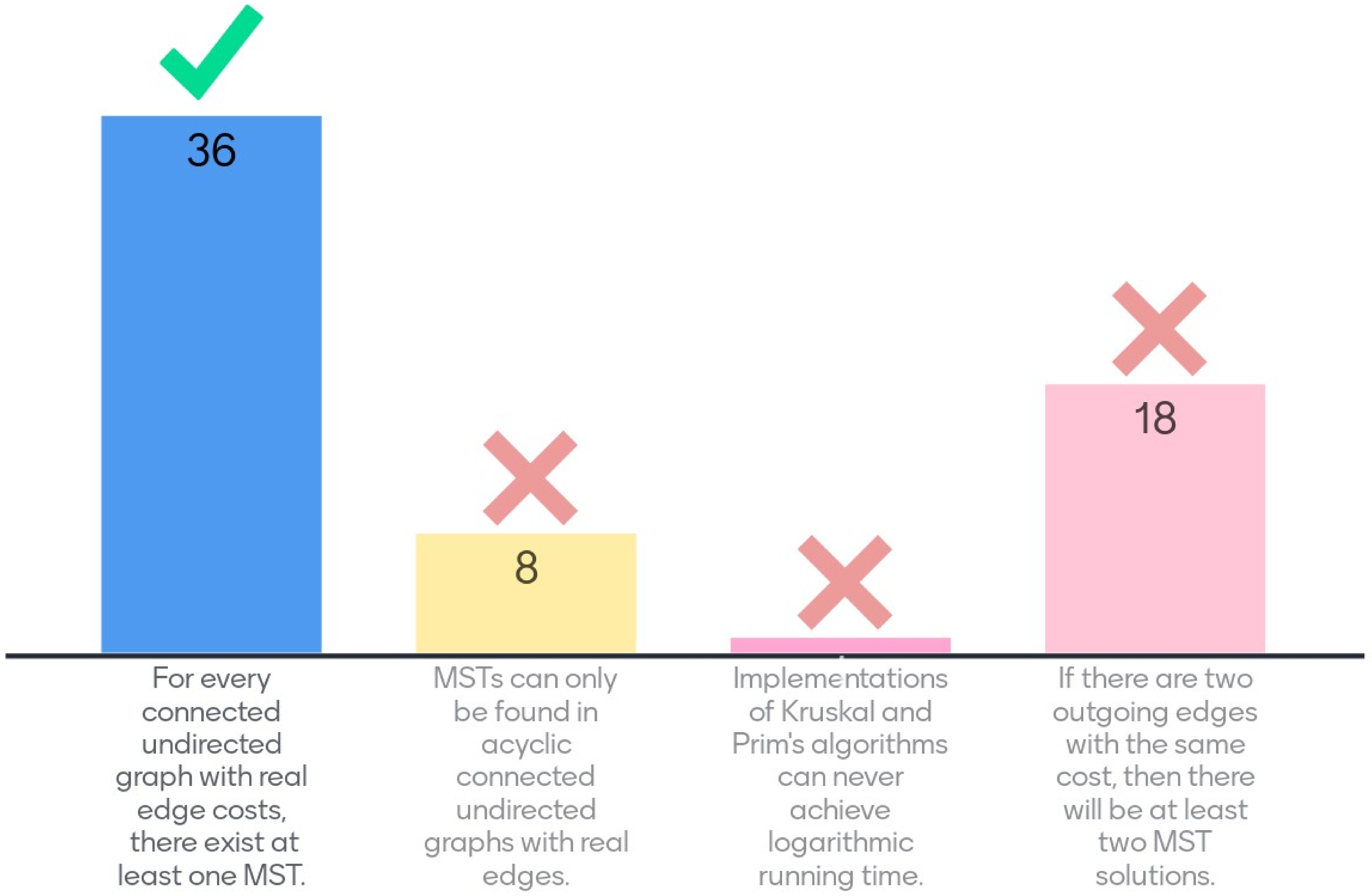
# Divide and conquer?



**Mark all answers that are TRUE about the minimum spanning tree problem.**

- [ ] Minimum spanning trees can only be found in acyclic connected undirected graphs G = (V, E) with real-valued edge costs.
- [ ] If there are two outgoing edges with the same cost, then there will be at least two minimum spanning tree solutions.
- [ ] For every connected undirected graph G = (V, E) with real-valued edge costs, there exist at least one minimum spanning tree.
- [ ] Implementations of Kruskal and Prim's algorithms can never achieve logarithmic running time.

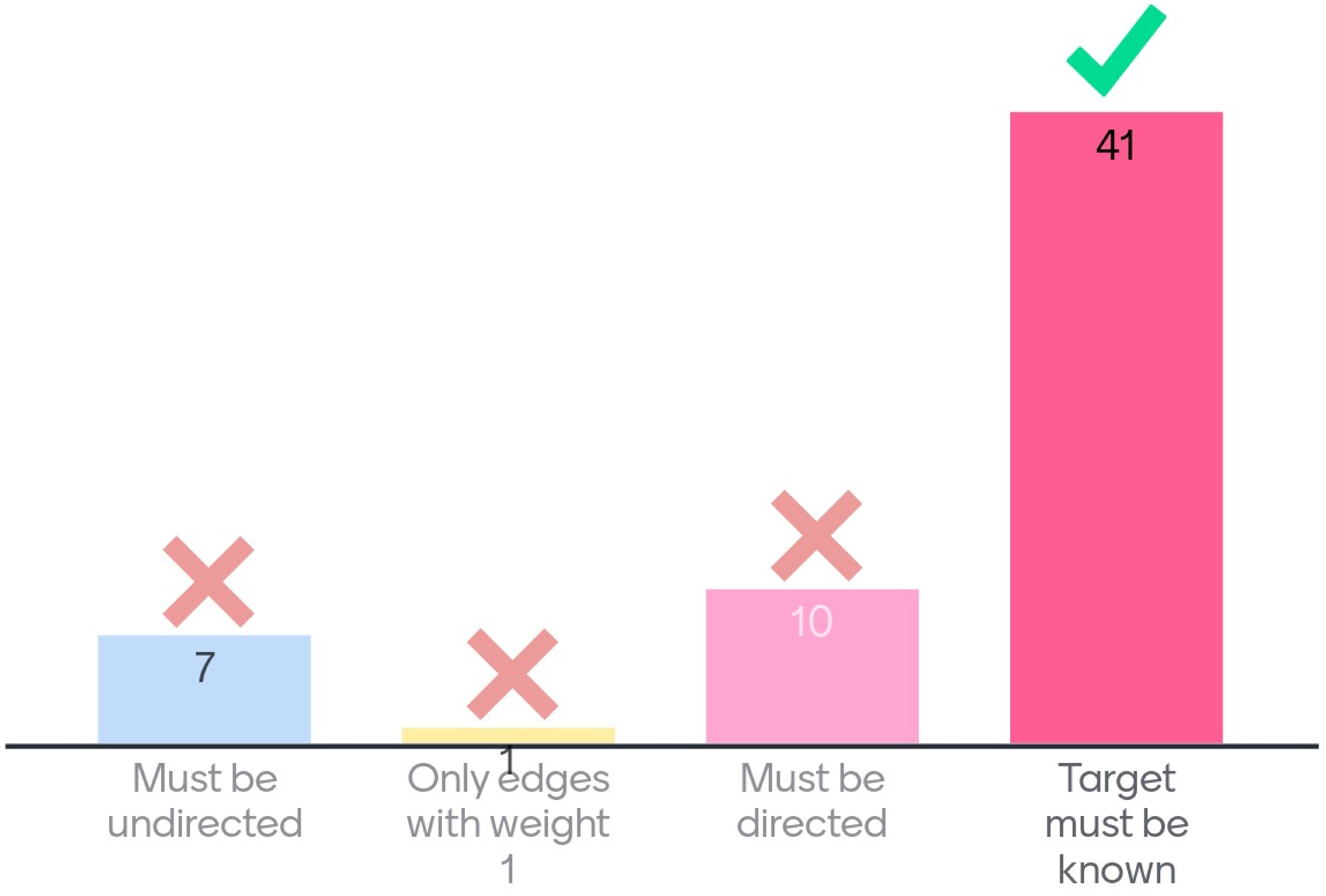
# Select answer(s) that are TRUE



**Which of the assumptions below have to hold for the A* algorithm to correctly find a shortest path from a single source vertex to a single target vertex?**

- [ ] The graph must contain only edges with weight 1.
- [ ] The graph must be directed.
- [ ] Target vertex must be known.
- [ ] The graph must be undirected.

# Which assumption must hold?

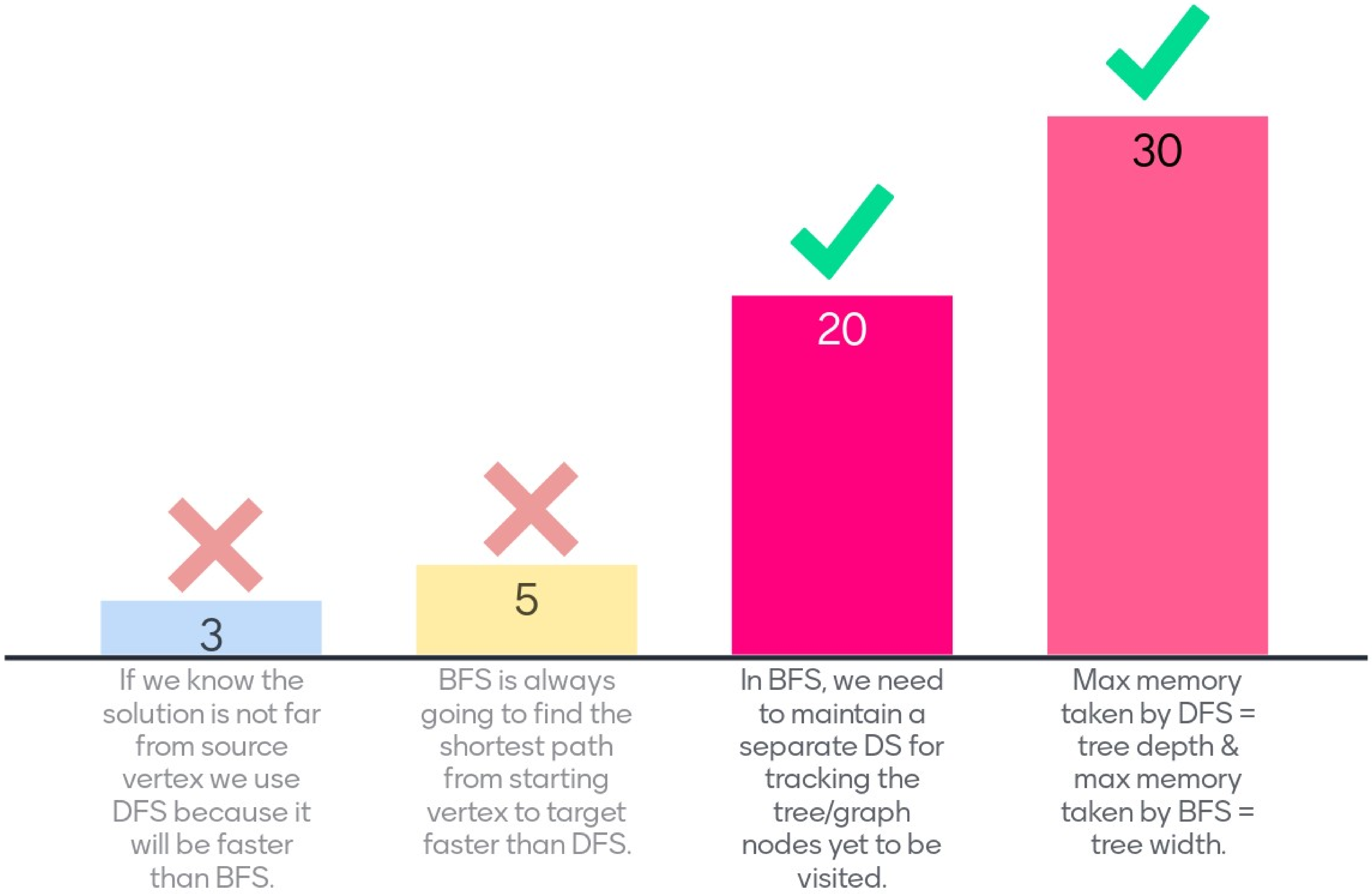


**Select the statements that are TRUE about graph search algorithms.**

- [ ] Breadth-first search is always going to find the shortest path from starting vertex s to target t faster compared to depth-first search.
- [ ] The maximum memory taken by depth-first search is equal to the depth of the tree, and the maximum memory taken by breadth-first search is equal to the width of the tree.
- [ ] If we know the solution is not that far from the source vertex we use depth-first search because it will find the solution faster compared to breadth-first search.
- [ ] In breadth-first search, we need to maintain a separate data structure for tracking the tree/graph nodes yet to be visited.

12

# Select answer(s) that are TRUE




23

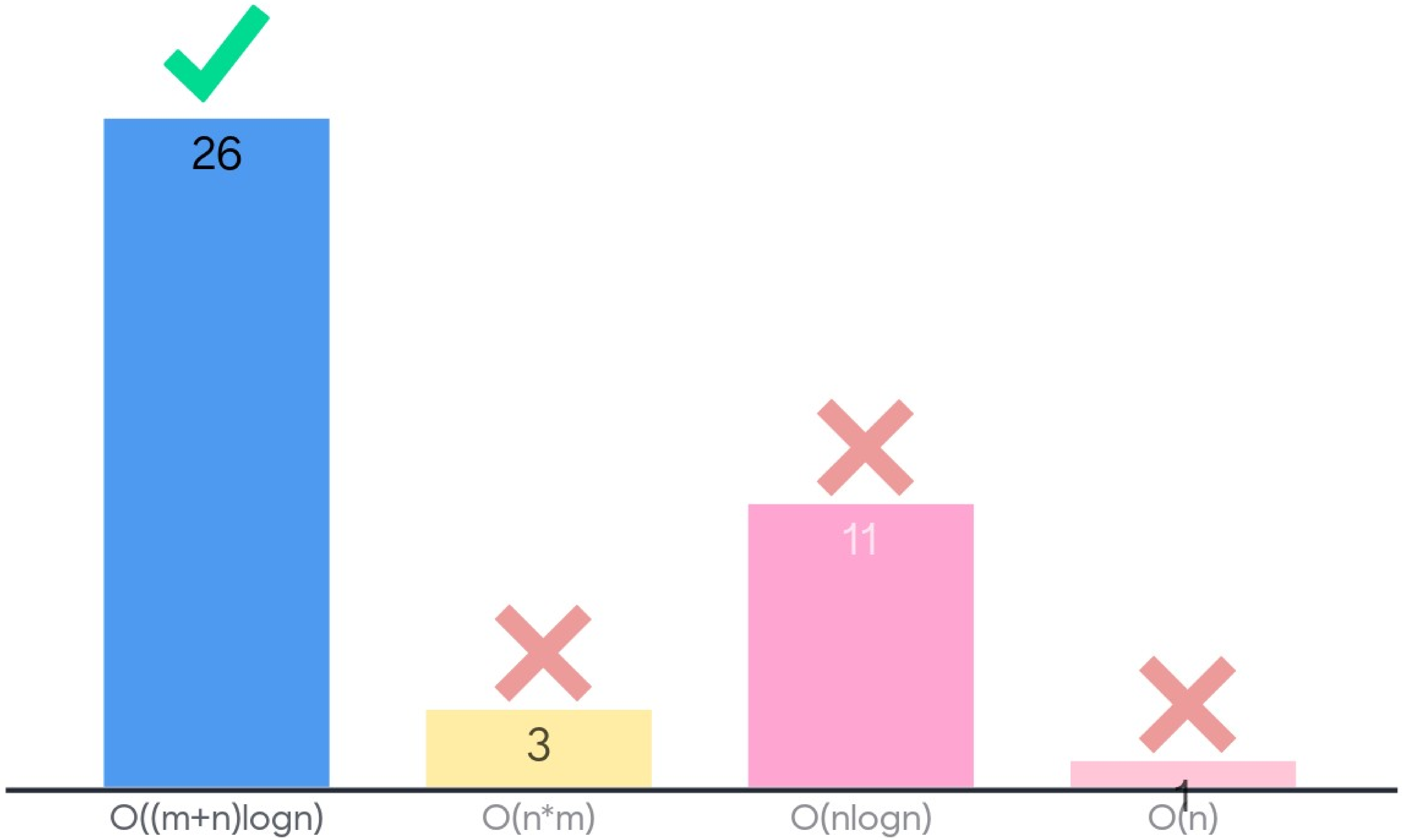
**What is the complexity of the Dijkstra algorithm that uses a heap data structure to store the yet unprocesses nodes (in terms of Big-O notation)?**
**(n denotes the number of nodes, while m denotes the number of edges in a directed, weighted graph with no negative weights. The key of the heap is the minimum Dijkstra score.)**

---

- [ ] O(nlogn)
- [ ] O((m+n)logn)
- [ ] O(n)
- [ ] O(n*m)

4

# What's the complexity?

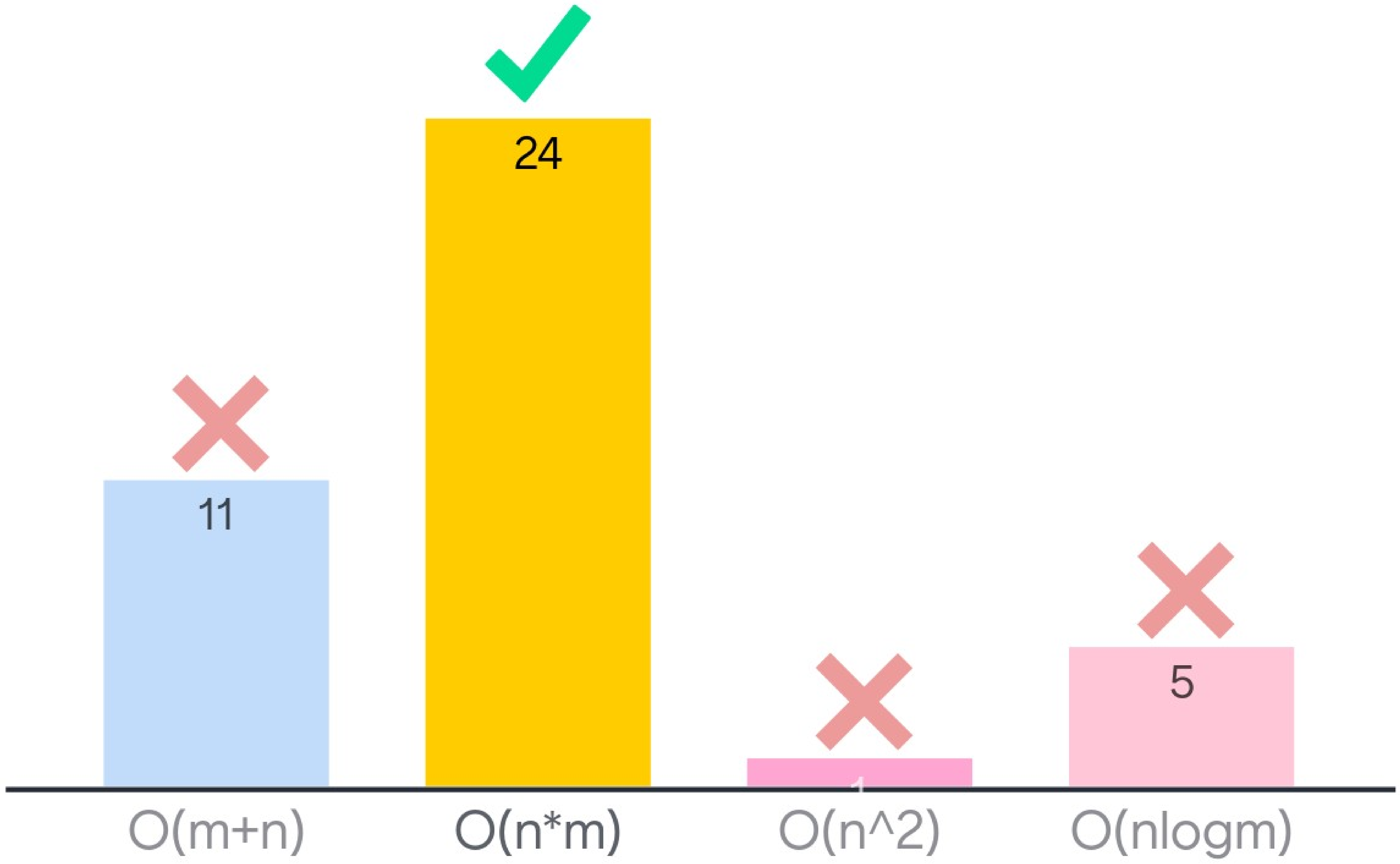


**What is the Big-O complexity (expressed in terms of n (|V|) and m (|E|)) of a straight forwards implementation (without using particular data structure) of Prim's algorithm?**

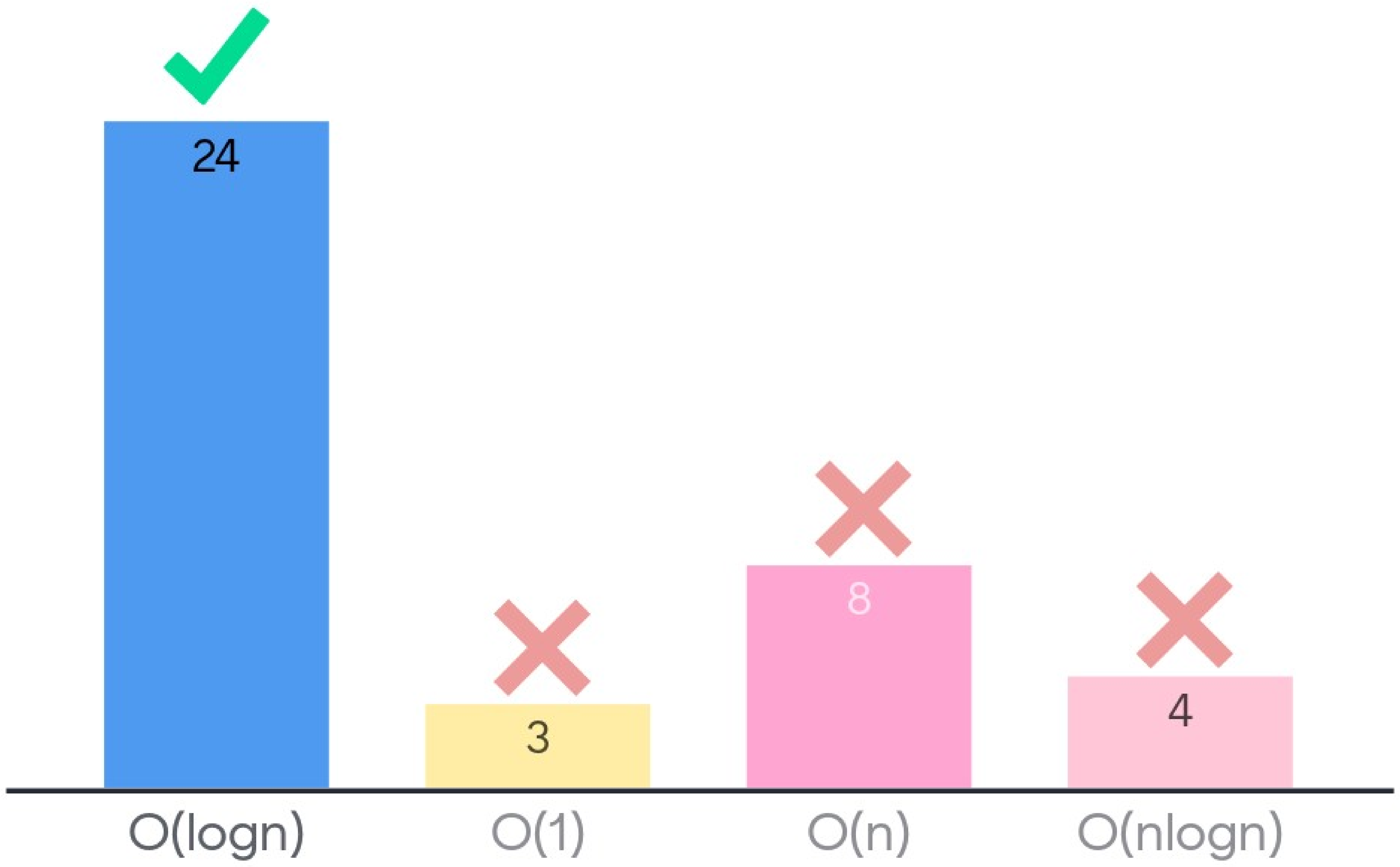
- [ ] O(n^2)
- [ ] O(m*n)
- [ ] O(m+n)
- [ ] O(nlog(m))

2

# What's the complexity?

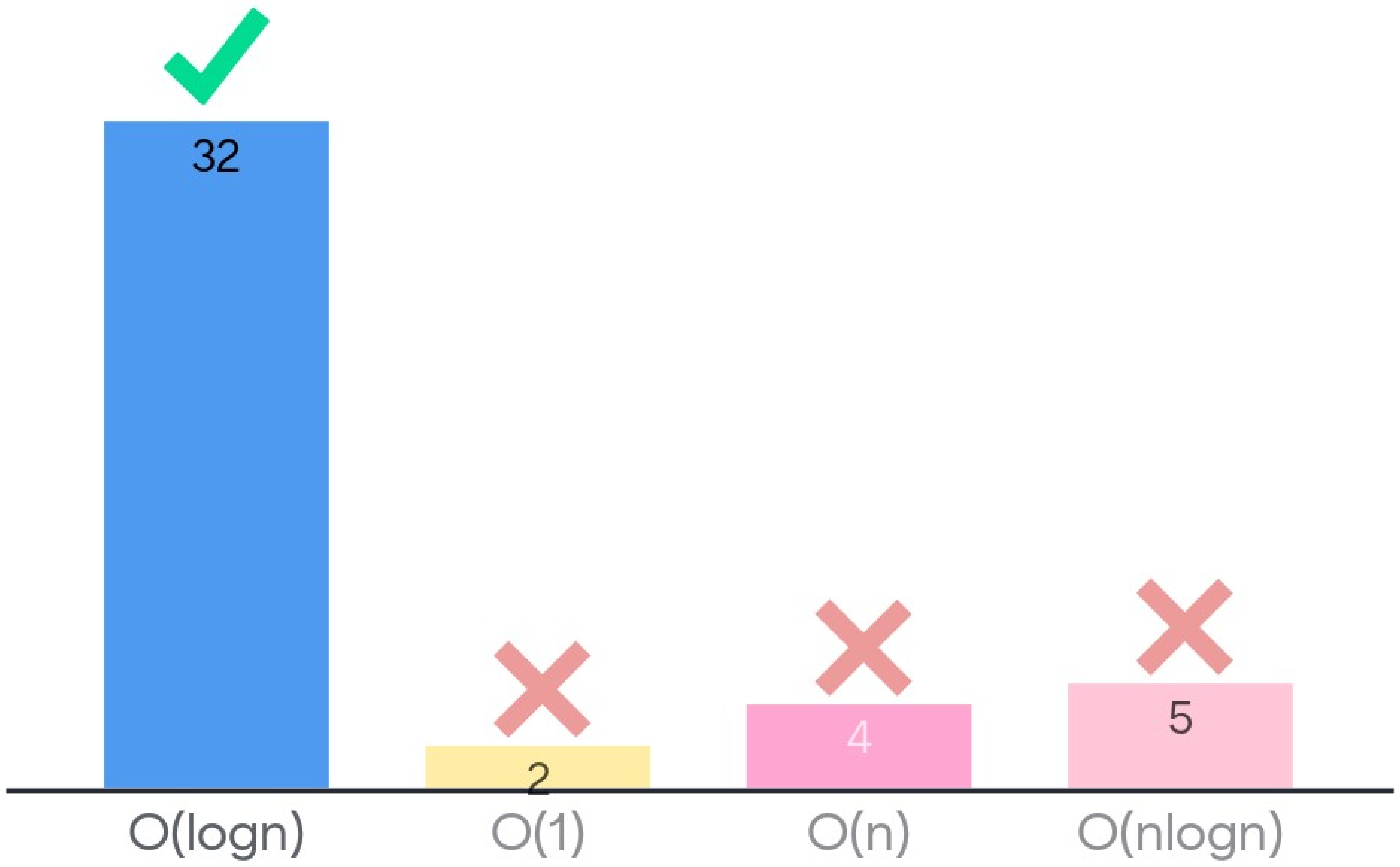


# What is the running time (in terms of Big-O) of the operations complexity for the heap data structure? DELETE



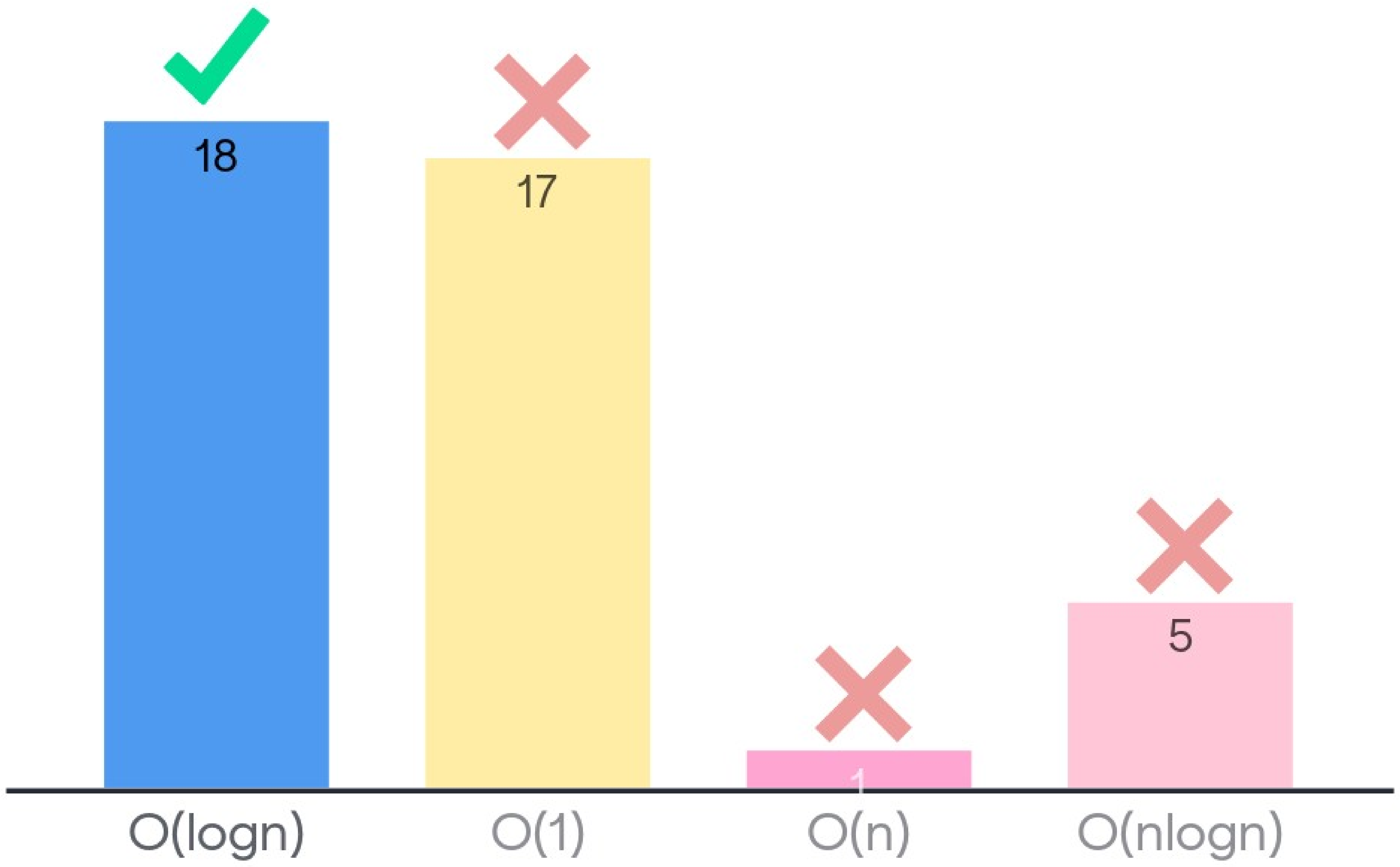

5

# What is the running time (in terms of Big-O) of the operations complexity for the heap data structure? INSERT



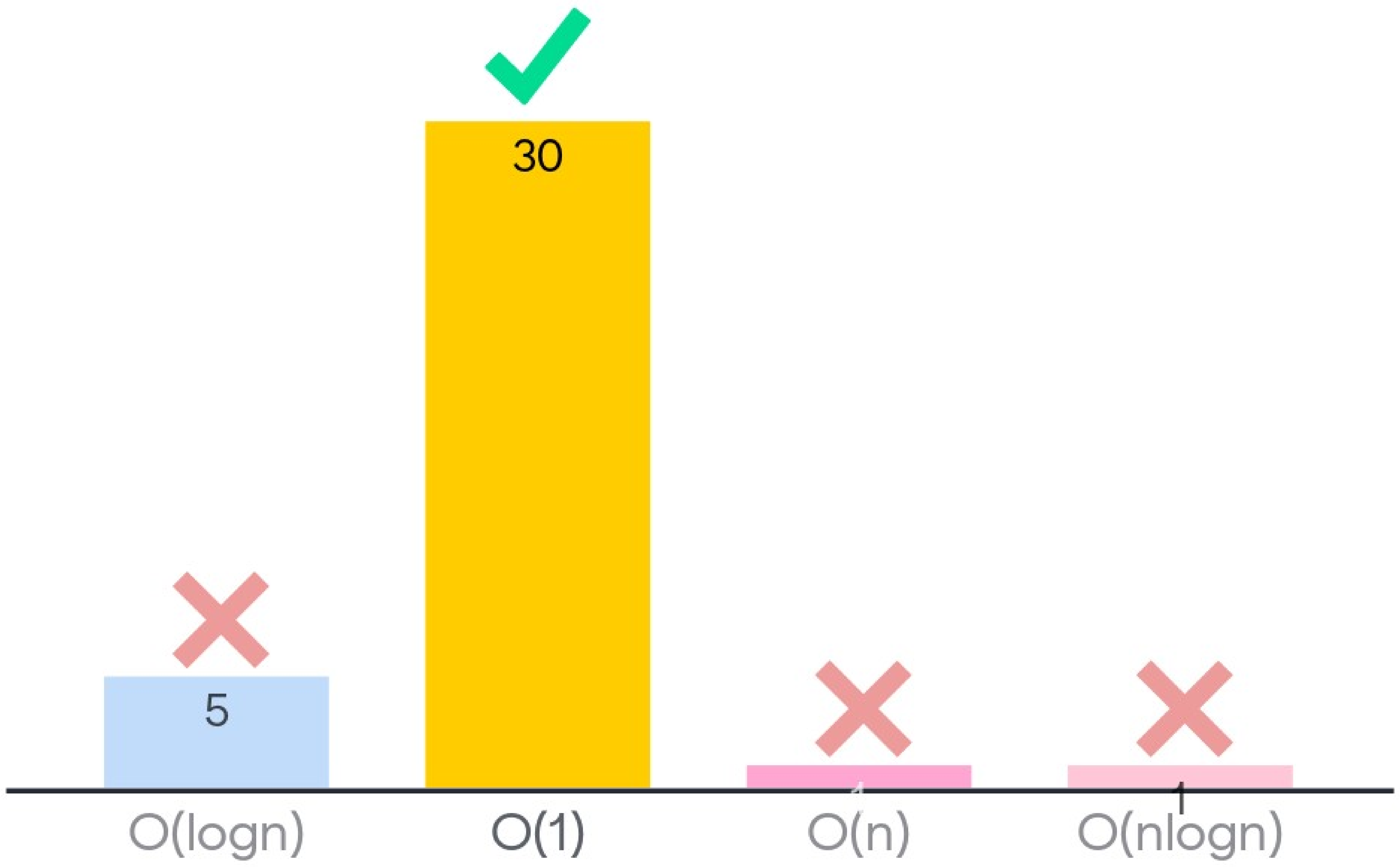

11

# What is the running time (in terms of Big-O) of the operations complexity for the heap data structure? EXTRACT MIN

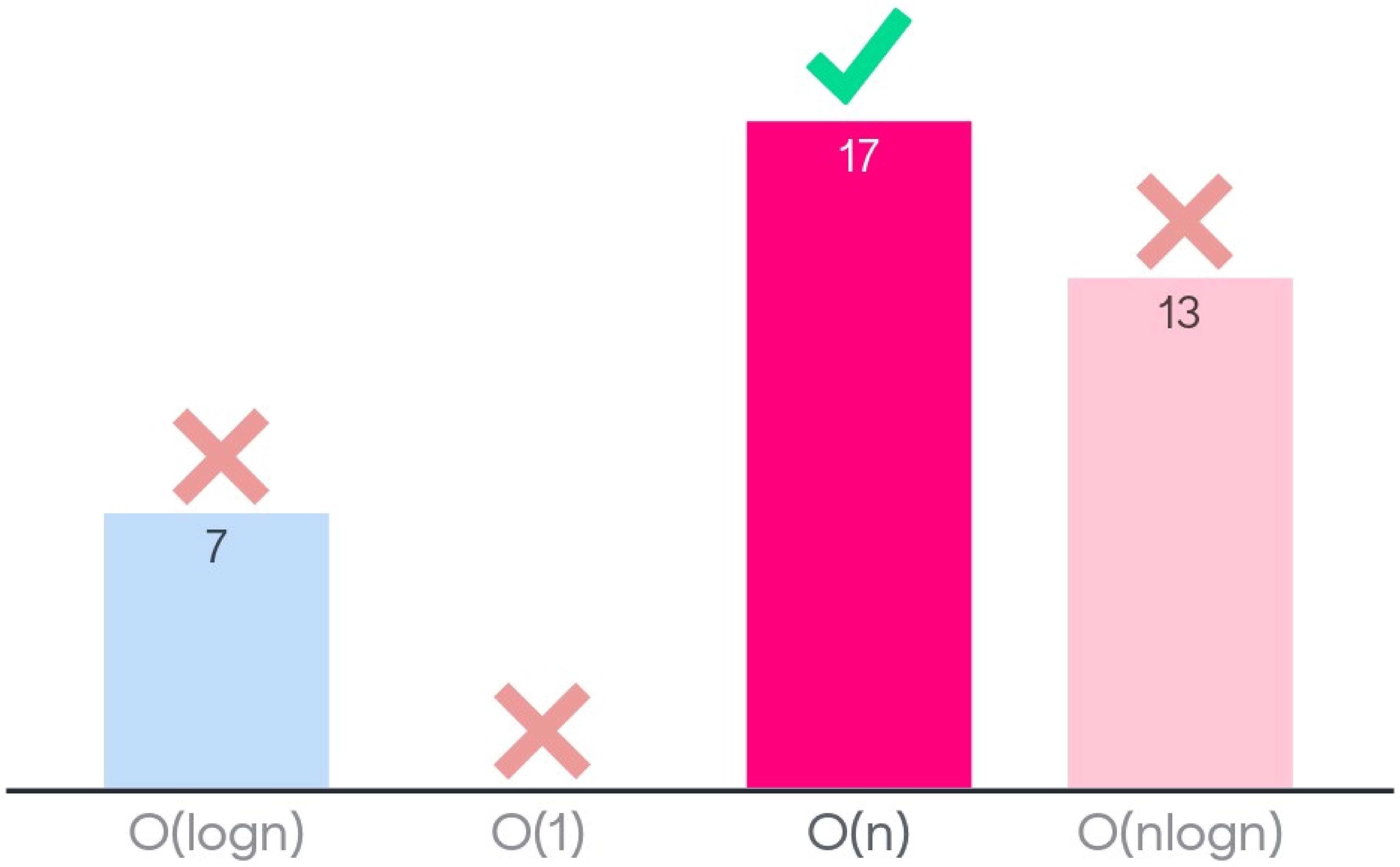



14

# What is the running time (in terms of Big-O) of the operations complexity for the heap data structure?
# FIND MIN



# What is the running time (in terms of Big-O) of the operations complexity for the heap data structure? HEAPIFY




5