Rosa Ekström
Linnéuniversitetet, Kalmar
2018-10-12

Time Complexity - how a algorithm scales

nodes = N
rootIterator = R (bad name now after I see)

Depth-First Search as implemented in X.MyDFS.dfs(DirectedGraph<E> graph)

```
while-loop  = O(N)
        if-sats = O(1)
                dfs() = O(R)
```

worst case scenario: all nodes are head nodes O(2N+R)
Solution
O(N+R)

Breadth-First Search as implemented in X.MyBFS.bfs(DirectedGraph<E> graph)

```
while-loop  = O(N)
        if-sats = O(1)
                bfs() = O(R)
```

Solution:
O(R+N)

Looks for the edges and previous ones.

Transitive Closure as implemented in X.MyTransitiveClosure.computeClosure(DirectedGraph<E> graph)

```
for-loop = O(N)
        while-loop = O(N)
                if-sats  = O(1)
                        compute = O(R)
```

Solution:

Rosa Ekström
Linnéuniversitetet, Kalmar
2018-10-12

O(N^2+NR)
Always compute each node that are visited.

Connected Components as implemented in X.MyConnectedComponents.computeComponents(DirectedGraph<E> graph)

```
while-loop = O(N)
    if-sats  = O(1)
        dfs = O(R)
            for-loop = O(N)
                if-sats  = O(1)
                    for-loop = O(N)
                        if-sats = O(1)
                            addAll()
                            addAll()
```

Solution:
O(N(N+R + N(N(2N)) → O(N(N(R+N + N^2*N) → O(N^2 + RN + N^4)
Since it's two addAll it equals 2N.
Since its two if-sats but only one can loop through it so is it either one of the if-sats.
Else it would be 4N

Inspiration:
https://www.youtube.com/watch?v=9cqGpAPqKkc (lesson 7 in 2v600)
https://www.youtube.com/watch?v=V6mKVRU1evU (2018-10-10)
https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/ (2018-10-12)