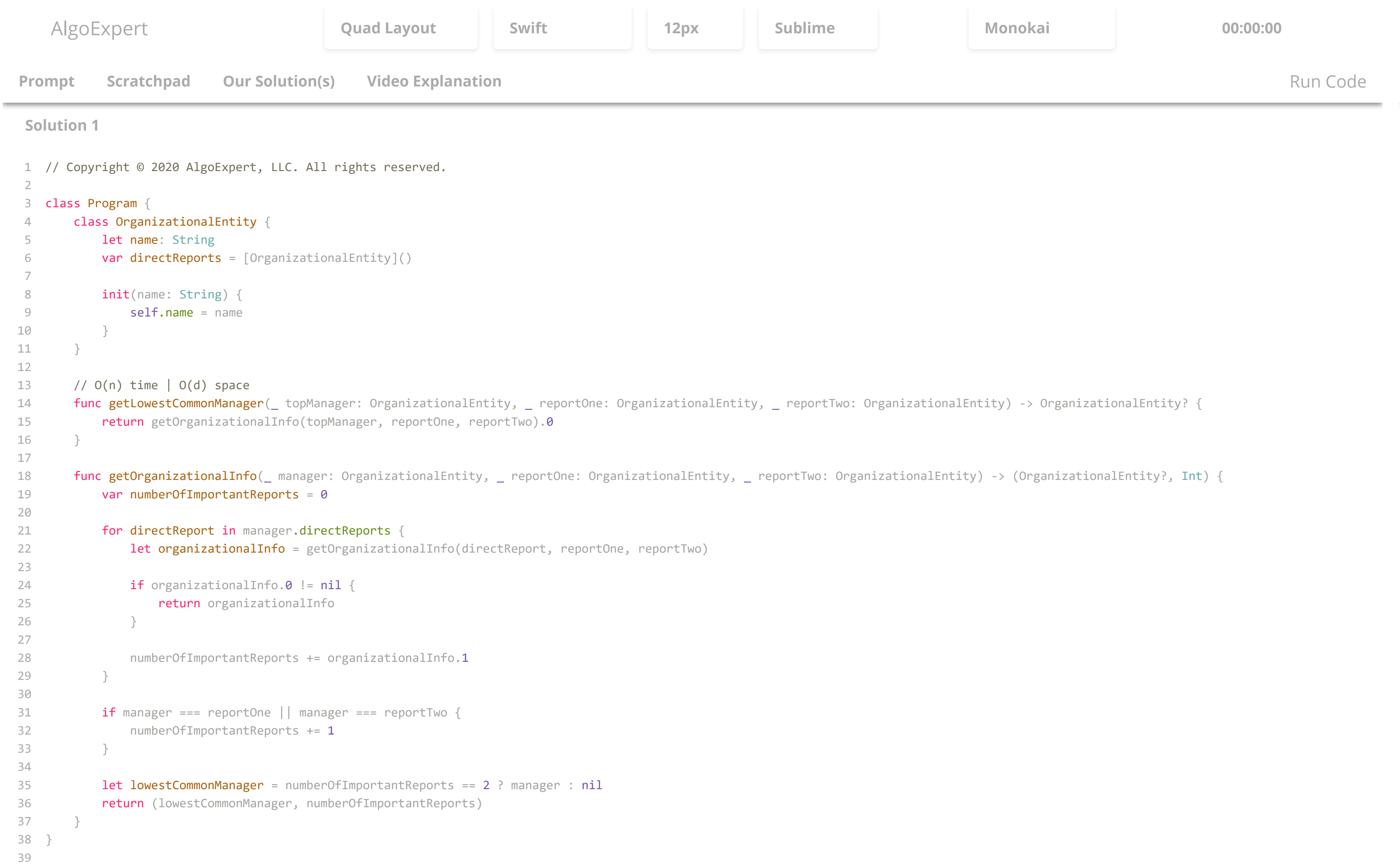

Solution 1

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    class OrganizationalEntity {
        let name: String
        var directReports = [OrganizationalEntity]()

        init(name: String) {
            self.name = name
        }
    }

    // O(n) time | O(d) space
    func getLowestCommonManager(_ topManager: OrganizationalEntity, _ reportOne: OrganizationalEntity, _ reportTwo: OrganizationalEntity) -> OrganizationalEntity? {
        return getOrganizationalInfo(topManager, reportOne, reportTwo).0
    }

    func getOrganizationalInfo(_ manager: OrganizationalEntity, _ reportOne: OrganizationalEntity, _ reportTwo: OrganizationalEntity) -> (OrganizationalEntity?, Int) {
        var numberOfImportantReports = 0

        for directReport in manager.directReports {
            let organizationalInfo = getOrganizationalInfo(directReport, reportOne, reportTwo)

            if organizationalInfo.0 != nil {
                return organizationalInfo
            }

            numberOfImportantReports += organizationalInfo.1
        }

        if manager === reportOne || manager === reportTwo {
            numberOfImportantReports += 1
        }

        let lowestCommonManager = numberOfImportantReports == 2 ? manager : nil
        return (lowestCommonManager, numberOfImportantReports)
    }
}
```