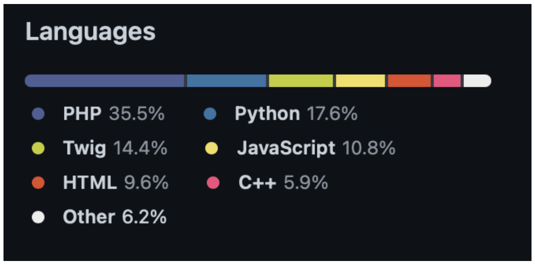
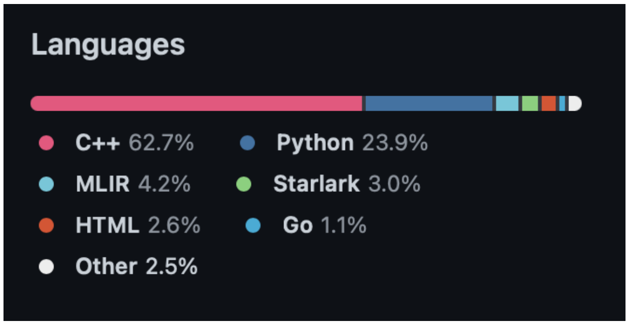
Joey Hinckley
661988002
This assignment can be found at:
https://github.com/JoeyHinckley34/oss-repo-template/blob/master/Assignments/Analysis%20Homework/Analysis%20of%20an%20Open%20Source%20Project.md

Project 1: Submitty https://github.com/Submitty/Submitty

Evaluation Factor	Level (0-2)	Evaluation Data
Licensing	2	BSD 3-Clause License. A permissive license (like MIT License) that forbids the use of the name of any of the developers for the promotion of derived versions without written consent. https://github.com/Submitty/Submitty/blob/master/LICENSE.md https://choosealicense.com/licenses/bsd-3-clause/
Language	1	 Languages ● PHP 35.5% ● Python 17.6% ● Twig 14.4% ● JavaScript 10.8% ● HTML 9.6% ● C++ 5.9% ● Other 6.2% I am unfamiliar with PHP and Twig. Also, I am hoping to learn Swift this semester, which is not present in this project.
Level of Activity	2	The only week in the past year without a commit was the week of 10/17/21. Overall very active.
Number of Contributors	2	148 Contributors total and 27 listed as still active. https://github.com/Submitty/Submitty/blob/master/AUTHORS.md
Product Size	1	Running the command: git ls-files -z | xargs -0 wc -l; Yields 756,092 total lines, which is a large codebase
Issue Tracker	2	246 issues open and 2,290 closed. The most recent issue opened 3 days ago, and the most recent issue closed 15 days ago.
New Contributor	2	There is a whole page dedicated to new developers coming in. https://submitty.org/developer
Community Norms	2	There is a well-put-together website to welcome users as well as developers. https://submitty.org/index/overview
User Base	2	I do not need to look far to find many users using this project, as I have used it every year at RPI and many of my peers have as well. The project also has 293 forks and 348 stars which shows quite a large user base.
Total Score	16	Overall this is an amazing open-source project, with an easy and clear guide for developers to come aboard. I just do not know yet if it is the project for me.

Project 2: Tensor Flow https://github.com/tensorflow/tensorflow

Evaluation Factor	Level (0-2)	Evaluation Data
Licensing	2	Apache License 2.0. A permissive license that calls for the copyright and license notice to be preserved in future versions. https://github.com/tensorflow/tensorflow/blob/master/LICENSE https://choosealicense.com/licenses/apache-2.0/
Language	1	 Languages ● C++ 62.7% ● Python 23.9% ● MLIR 4.2% ● Starlark 3.0% ● HTML 2.6% ● Go 1.1% ● Other 2.5% Despite being very familiar with C++ and Python, this still is not the best fit. I want to use this semester as an opportunity to learn Swift which is not present in this project.
Level of Activity	2	A massive number of commits a week. Averaging 300+ commits every week with the lowest in the past year being 184.
Number of Contributors	2	There are 3062 total contributors listed on Github, which is a very large community of contributors.
Product Size	1	Running the command: git ls-files -z | xargs -0 wc -l; Yields 6 different totals which in total sum to 5,849,416, which is a massive codebase.
Issue Tracker	2	There are 2,340 open issues and 31,642 closed. The most recent open was 1 hour ago and the most recent closed was 11 hours ago. Just about as active as you can be.
New Contributor	2	There is a tab on their website dedicated to welcoming new contributors. https://www.tensorflow.org/community/contribute/code
Community Norms	2	There is a professional website dedicated to this project. https://www.tensorflow.org
User Base	2	There are 176,063 users listed on github, also 86.3k forks and 163k stars. This shows just how wide spread this project is
Total Score	16	This project is far too big for me to become a part of. It will take me a while to understand how to use tensorflow, and even long to be able to contribute any meaningful feature to the source code.

Project 3: Swift Algorithm Club https://github.com/raywenderlich/swift-algorithm-club

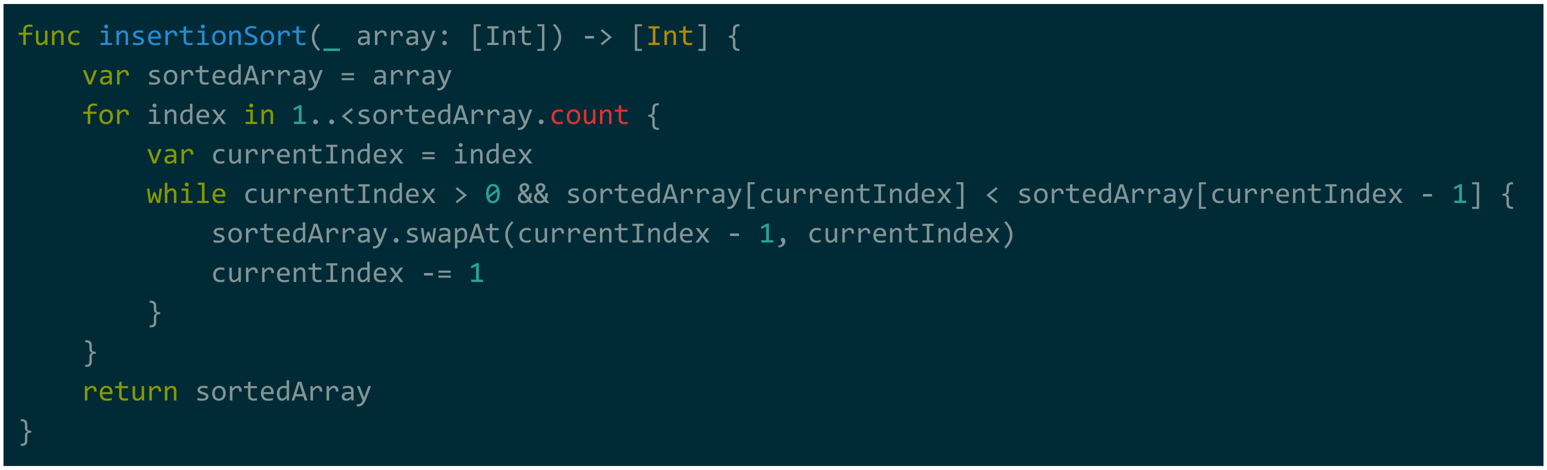
Evaluation Factor	Level (0-2)	Evaluation Data
Licensing	2	MIT License. A permissive lisence which only stipulation is that the copyright and license notice be included in any projects which use a significant portion of the code. https://github.com/raywenderlich/swift-algorithm-club/blob/master/LICENSE.txt
Language	2	Languages ● Swift 99.5% ● Other 0.5% This project is entirely Swift based and is intended to be educational, which is exactly what I am looking for.
Level of Activity	1	Not nearly as active as the other two projects, with just 5 commits in 2021.
Number of Contributors	2	There are 256 contributors listed on Github, which shows a well established community.
Product Size	2	Running the command: git ls-files -z | xargs -0 wc -l; Yields 276,358 total lines, which is much more manageable of a codebase when compared to the other two projects.
Issue Tracker	2	Only have 26 issues open and 180 closed. The most recent open was 5 days ago and the most recent closed was Jan 5, 2022. Which is not nearly as active as the other two projects, but not “very low or sporadic” enough to warrant a 1.
New Contributor	2	There is a file the repo dedicated to guiding developers through the process of contributing. https://github.com/raywenderlich/swift-algorithm-club/blob/master/.github/CONTRIBUTING.md
Community Norms	2	There is a website dedicated to the project. https://aquarchitect.github.io/swift-algorithm-club/
User Base	2	There are 4.7k forks and 25.9k stars on the repo in github, which shows this project is quite popular, as well as having a whole book associated with the topic. https://www.raywenderlich.com/books/data-structures-algorithms-in-swift
Total Score	17	This project is not nearly as active as the other two I researched, it is much more in line with what I want to learn this semester. Not only is this repository in the language I am looking for, but also it was made for educational purposes.

An In Depth Analysis of Swift Algorithm Club

Going into this semester, I wanted to learn a new language and decided that Swift would be that language. I did not want to limit myself by only choosing projects that were in Swift to research, so I picked my favorite Swift project and two others that were drastically different. Of the three projects chosen, Submitty, TensorFlow, and Swift Algorithm Club, the former two received the same sixteen points on the rubric given by the foss2serve rubric. Submitty and TensorFlow both lost points for the same reason. They were too big and in a language I either am unfamiliar with or do not want to work with this semester. On the other hand, Swift Algorithm Club only lost a point for being relatively inactive in the past year. Due to Swift Algorithm Club receiving the highest score as well as being in the language I would like to learn, I chose it for my in-depth analysis.

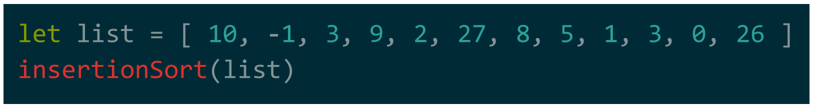
The first thing I had to do when choosing a project was to ensure there was a license for it guaranteeing it to be either free or open source. Swift Algorithm Club has the MIT License as their chosen license. This license is the standard permissive license. It allows the user nearly full rein to use the software however they please. Since it is a permissive license, it allows for the production and issue of closed source material. There are only two stipulations associated with the MIT License. The first being that if anyone uses the licensed software in a significant way, they are obligated to include the copyright notice and the license with their adapted code base. The second stipulation is there is a limitation on liability and no warranty. Since there is no warranty, there is no guarantee of the quality of the project. By no means does this mean the project was written poorly, but rather, the developers cannot be held accountable in a court of law if the code does not fully do what it is promised. The limitation on liability ensures that if this software or future iterations of it cause any damage, the developers cannot be found legally responsible. More information can be found here.

Swift Algorithm Club is a unique open source project in that it is completely centered around a single language. Most open source projects pull in multiple languages. Not only is this project completely centered around a language, it is set up to teach the language. This was optimal for me as I am hoping to learn Swift, so this was a helpful repository to find regardless of this assignment. For all their algorithms they have a detailed explanation of the algorithm itself and then how to implement such an algorithm in Swift. Looking at the simple insertion sort algorithm, they have an entire readme in markdown explaining in detail how the algorithm works. Starting off with two lists, unsorted and empty, then placing each element from the unsorted list into the empty list in order. Then it explains how to accomplish the same thing with a single list called an in-place sort. Then finally showing how the code works in Swift.



```
func insertionSort(_ array: [Int]) -> [Int] {
    var sortedArray = array
    for index in 1..<sortedArray.count {
        var currentIndex = index
        while currentIndex > 0 && sortedArray[currentIndex] < sortedArray[currentIndex - 1] {
            sortedArray.swapAt(currentIndex - 1, currentIndex)
            currentIndex -= 1
        }
    }
    return sortedArray
}
```

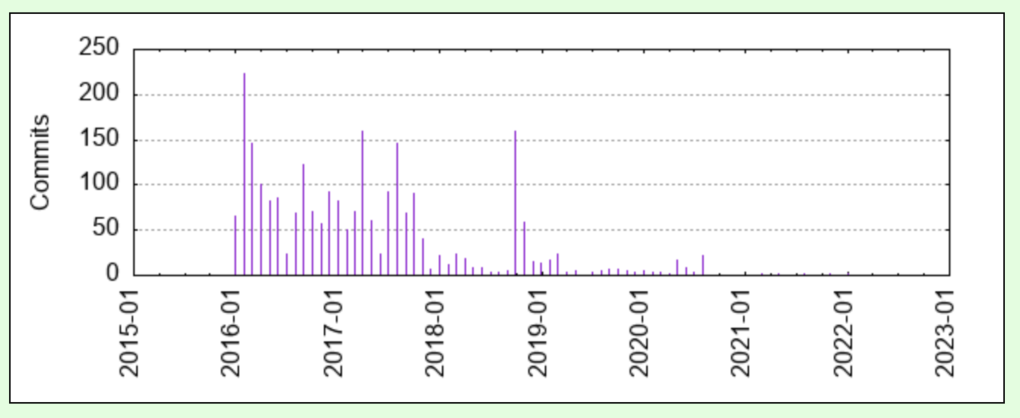
Insertion sort function implemented in Swift



```
let list = [ 10, -1, 3, 9, 2, 27, 8, 5, 1, 3, 0, 26 ]
insertionSort(list)
```

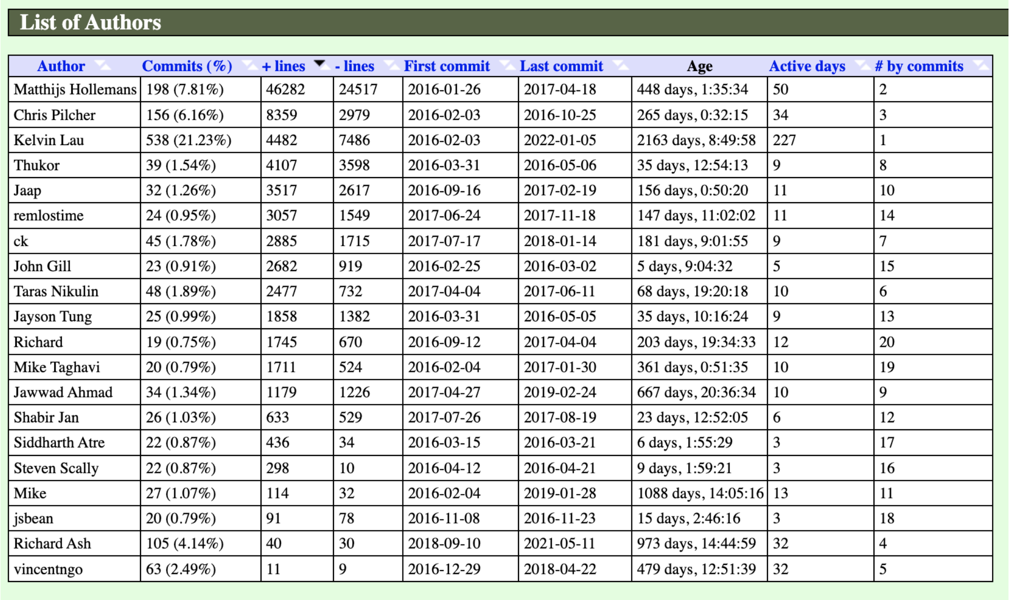
Calling insertion sort on a list

The level of depth is not just found in the beginner algorithms, all algorithms have a well-written readme explaining the algorithm as well as how it works in Swift. All in all, the knowledge available in the repository is similar to the knowledge base found in your everyday textbook. The reason behind this is that the owner of the repository, Ray Wenderlich, is an author himself. Ray Wenderlich is the founder of raywenderlich.com (go figure), which is a coding tutorial website. Not only is he the owner of a teaching website he has written a book on the subject called Data Structures and Algorithms in Swift.



All commits in repository history provided by GitStats

Despite how great this repository is, the only drawback is that it is no longer very popular, at least for developers. This chart shows a clear slow down in the amount of commits over time. The project started off very strongly in 2016, but since then it has lost its steam. By no means does this mean that this project is finished, more likely that its developers have just moved on to different things. This will not stop me or anyone from contributing as the manager of the repository still seems to be active.

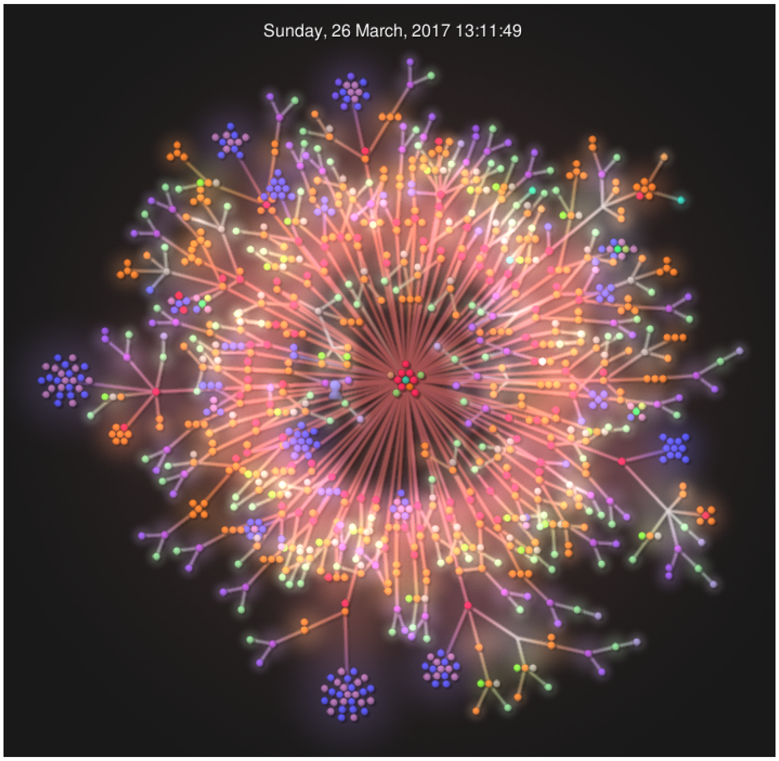


List of Authors

Author	Commits (%)	+ lines	- lines	First commit	Last commit	Age	Active days	# by commits
Matthijs Hollemans	198 (7.81%)	46282	24517	2016-01-26	2017-04-18	448 days, 1:35:34	50	2
Chris Pilcher	156 (6.16%)	8359	2979	2016-02-03	2016-10-25	265 days, 0:32:15	34	3
Kelvin Lau	538 (21.23%)	4482	7486	2016-02-03	2022-01-05	2163 days, 8:49:58	227	1
Thukor	39 (1.54%)	4107	3598	2016-03-31	2016-05-06	35 days, 12:54:13	9	8
Jaap	32 (1.26%)	3517	2617	2016-09-16	2017-02-19	156 days, 0:50:20	11	10
remlostime	24 (0.95%)	3057	1549	2017-06-24	2017-11-18	147 days, 11:02:02	11	14
ck	45 (1.78%)	2885	1715	2017-07-17	2018-01-14	181 days, 9:01:55	9	7
John Gill	23 (0.91%)	2682	919	2016-02-25	2016-03-02	5 days, 9:04:32	5	15
Taras Nikulin	48 (1.89%)	2477	732	2017-04-04	2017-06-11	68 days, 19:20:18	10	6
Jayson Tung	25 (0.99%)	1858	1382	2016-03-31	2016-05-05	35 days, 10:16:24	9	13
Richard	19 (0.75%)	1745	670	2016-09-12	2017-04-04	203 days, 19:34:33	12	20
Mike Taghavi	20 (0.79%)	1711	524	2016-02-04	2017-01-30	361 days, 0:51:35	10	19
Jawwad Ahmad	34 (1.34%)	1179	1226	2017-04-27	2019-02-24	667 days, 20:36:34	10	9
Shabir Jan	26 (1.03%)	633	529	2017-07-26	2017-08-19	23 days, 12:52:05	6	12
Siddharth Atre	22 (0.87%)	436	34	2016-03-15	2016-03-21	6 days, 1:55:29	3	17
Steven Scally	22 (0.87%)	298	10	2016-04-12	2016-04-21	9 days, 1:59:21	3	16
Mike	27 (1.07%)	114	32	2016-02-04	2019-01-28	1088 days, 14:05:16	13	11
jsbean	20 (0.79%)	91	78	2016-11-08	2016-11-23	15 days, 2:46:16	3	18
Richard Ash	105 (4.14%)	40	30	2018-09-10	2021-05-11	973 days, 14:44:59	32	4
vincentngo	63 (2.49%)	11	9	2016-12-29	2018-04-22	479 days, 12:51:39	32	5

Top twenty authors, ordered by lines of code committed

Of the top twenty top authors on the project fourteen of them had their last commit in 2017. While the data shows how the main contributors have long since left the project, Kelvin Lau seems to be keeping the repository alive. Although Kelvin has not written the most number of lines for the project, he more than doubles the number of commits as the developer who has the second most, Matthijs Hollemans. Since Kelvin has been active this year as well as handling pull requests in the past, it gives me great confidence that any contributions I may make will get added to the repository.


Sunday, 26 March, 2017 13:11:49

Running gource on the Repository.

The project is very secular. There is no overlap from one algorithm to another or one directory to another. Each folder is its own self contained piece of code with its description. This makes contributions extremely easy. All that would be needed would be finding an algorithm or data structure yet to be included and implementing it in Swift. Then explain that algorithm in layman's terms in a markdown file. One such data structure yet to be implemented is a Fibonacci Heap. The repository also has a number of puzzles such as the Monty Hall Problem and Fizz Buzz. Another possible route of contributing would be finding a new puzzle that can be implemented using Swift. The final way I can imagine contributing would be looking for errors and debugging the code that is already a part of the project. The reason this project took off was due to the fact that there are a multitude of paths a developer can take to contribute. This by no means a small project, but since everything is so separate from one another it is very manageable.

This project has a well kept issue tracker. The most common issues being topics someone wants improvements on or new topics to be implemented entirely. One example of this is the open issue #911 which is titled "Topics Wishlist." This issue was created by Kelvin linking to two other tickets, #735 and #876. #735 describes a Stack Machine as, "a type of computer that processes instructions by pushing and popping values to an internal 'stack.'" The issue then implements a rudimentary implementation of it in Slack with the issue asking for a more optimized solution. The other issue is asking for implementation of the Raft Consensus Algorithm, which is an algorithm used for servers communicating with each other. Other issues talk about possible optimization that can be made. For example #989, there is a discussion on optimizing the queue implementation. They go into the reasoning of why arrays were chosen as the data structure used in a queue rather than a linked list. The logic given in favor of arrays was that an array takes up a contiguous chunk of memory. This makes it better than a linked list because a linked lists nodes can be anywhere in memory and are connected through edges rather than being next to each other in memory. This gives linked lists a disadvantage as "it will cost more memory than using an array, because each node in LinkedList holds data and [a] reference to next. Meanwhile, an array holds only actual data and its index."

Over the course of this project's life it has garnered immense popularity. On github it has over twenty five thousand stars and almost five thousand forks. Even though this project has fallen out of the main light in recent years, its two hundred and fifty six contributors show how this is a successful open source project. Swift Algorithms Club has built its own community around implementing the most well known topics in Swift as well as giving in-depth analysis of them all.