

哈尔滨工业大学 计算学部

2023年秋季学期《开源软件开发实践》

Lab 2：开源软件开发协作流程

姓名	学号	联系方式
李璉峻	2021130014	kohakerfle@naver.com /17645005280

目 录

1 实验要求.....	1
2 实验内容1 发送pull request	1
2.1 fork项目	1
2.2 git操作命令	1
2.3 代码修改.....	1
2.4 测试通过截图.....	1
3 实验内容2 接受pull request	1
4 实验内容3 github辅助工具	1
4.1 熟悉GoodFirstIssue工具	1
4.2 安装并使用Hypercrx	2
4.3 利用OpenLeaderboard工具	2
5 小结.....	3

[文档全部完成之后，请更新上述区域]

1 实验要求

简要复述实验手册中要求达到的实验目标与过程。

这次实验的目标是：

了解和掌握基于代码托管平台的开源软件协作开发过程。

掌握基于github的软件项目协作开发命令和方法。

熟悉几个github中常用开源软件开发工具。

2 实验内容1 发送pull request

2.1 fork项目

给出fork后的个人仓库中项目的界面截图

```
83
84  ✓ private long gcd(long a, long b) {
85         long remainder = a % b;
86         while (remainder != 0) {
87             a = b;
88             b = remainder;
89             remainder = a % b;
90         }
91         return b;
92     }
93 }
```

2.2 git操作命令

给出本地clone、提交、远程推送等步骤的git命令和对应的截图

```
MINGW64:/c/Lab1/my_project
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git clone "https://github.com/lee-yeonjoon/OSSDP-Lab2.git"
Cloning into 'OSSDP-Lab2'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 30 (delta 2), reused 1 (delta 1), pack-reused 22
Receiving objects: 100% (30/30), 22.32 KiB | 105.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
```

\$ git clone “<https://github.com/lee-yeonjoon/OSSDP-Lab2>”

```
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git checkout -b fix-branch
fatal: a branch named 'fix-branch' already exists
```

\$ git checkout -b fix-branch

```
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git add .
warning: adding embedded git repository: Lab2
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:   git submodule add <url> Lab2
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:   git rm --cached Lab2
hint: See "git help submodule" for more information.
warning: adding embedded git repository: OSSDP-Lab2
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ echo "Lab2/">> .gitignore
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git commit -m "Fix:Fixing errors and adding tests"
[fix-branch e98f8c8] Fix:Fixing errors and adding tests
3 files changed, 3 insertions(+)
create mode 100644 .gitignore
create mode 160000 Lab2
create mode 160000 OSSDP-Lab2
```



\$ git add .

\$ git commit -m "Fix:Fixing errors and adding tests"


```
kohak@LAPTOP-TROBMSUK MINGW64 /c/Lab1/my_project (fix-branch)
$ git push origin fix-branch
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (15/15), 1.42 KiB | 725.00 KiB/s, done.
Total 15 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
remote:
remote: Create a pull request for 'fix-branch' on GitHub by visiting:
remote:   https://github.com/lee-yeonjoon/Lab1/pull/new/fix-branch
remote:
To https://github.com/lee-yeonjoon/Lab1.git
 * [new branch]      fix-branch -> fix-branch
```

\$ git push origin fix-branch


Solution10 修改错误 #48


 lee-yeonjoon wants to merge 2 commits into `HanchuanXu:main` from `lee-yeonjoon:main` 



Conversation 0 Commits 2 Checks 0 Files changed 1

 lee-yeonjoon commented now


修改错误



 lee-yeonjoon added 2 commits yesterday

-  Update Solution10.java Verified a62d92f
-  Update Solution10.java Verified 0efa2f1

Add more commits by pushing to the `main` branch on [lee-yeonjoon/OSSDP-Lab2](#).

 This branch has no conflicts with the base branch
Only those with [write access](#) to this repository can merge pull requests.

[HanchuanXu/OSSDP-Lab2/pull/48/checks](#)

2.3 代码修改

将修改完善后的原文件粘贴在此处

```

/*solution10*/
/**
 * @description:
 *      给定一个表示分数加减运算的字符串 expression ，你需要返回一个字符串形式的计算结果。
 *
 *      这个结果应该是不可约分的分数，即最简分数。 如果最终结果是一个整数，例如 2，你需要将它转换成分数形式，其分母为
 *      1。所以在上述例子中，2 应该被转换为 2/1。
 *
 *
 *
 *
 *
 *
 *      示例 1:
 *
 *      输入: expression = "-1/2+1/2"
 *      输出: "0/1"
 *
 *      示例 2:
 *
 *      输入: expression = "-1/2+1/2+1/3"
 *      输出: "1/3"
 *
 *      示例 3:
 *
 *      输入: expression = "1/3-1/2"
 *      输出: "-1/6"
 *
 *
 *
 *
 *
 *
 *      提示:
 *
 *      输入和输出字符串只包含 '0' 到 '9' 的数字，以及 '/', '+' 和 '-'。
 *      输入和输出分数格式均为 ±分子/分母。如果输入的第一个分数或者输出的分数是正数，则 '+' 会被省略掉。
 *      输入只包含合法的最简分数，每个分数的分子与分母的范围是 [1,10]。如果分母是1，意味着这个分数实际上是一个整数。
 *      输入的分数个数范围是 [1,10]。
 *      最终结果的分子与分母保证是 32 位整数范围的有效整数。
 */
public class Solution10Test2021130014 {
    public static void main(String[] args) {
        Solution10 solution = new Solution10();
        String expression = "-1/2+1/2"; // 要测试的表达式
        String result = solution.calculateFraction(expression);
        System.out.println("Result: " + result);
    }
}

```

```
    }  
}  
  
class Solution10 {  
    public String calculateFraction(String expression) {  
        long numerator = 0, denominator = 1; // 分子, 分母  
        int index = 0, n = expression.length();  
  
        while (index < n) {  
            // 读取分子  
            long numerator1 = 0, sign = 1;  
            if (expression.charAt(index) == '-' || expression.charAt(index) == '+') {  
                sign = expression.charAt(index) == '+' ? 1 : -1;  
                index++;  
            }  
  
            while (index < n && Character.isDigit(expression.charAt(index))) {  
                numerator1 = numerator1 * 10 + expression.charAt(index) - '0';  
                index++;  
            }  
  
            numerator1 *= sign;  
            index++;  
  
            // 读取分母  
            long denominator1 = 1;  
            if (index < n && expression.charAt(index) == '/') {  
                index++;  
                while (index < n && Character.isDigit(expression.charAt(index))) {  
                    denominator1 = denominator1 * 10 + expression.charAt(index) - '0';  
                    index++;  
                }  
            }  
  
            denominator1 = Math.abs(denominator1);  
  
            numerator = numerator * denominator1 + numerator1 * denominator;  
            denominator *= denominator1;  
  
            long gcd = gcd(Math.abs(numerator), denominator);  
            numerator /= gcd;  
            denominator /= gcd; // 使用最小公倍数转换为最小化分数  
        }  
    }  
}
```

```
        if (numerator % denominator == 0) {
            return Long.toString(numerator / denominator) + "/1"; // 结果为整数
        }

        if (numerator == 0) {
            return "0/1";
        }

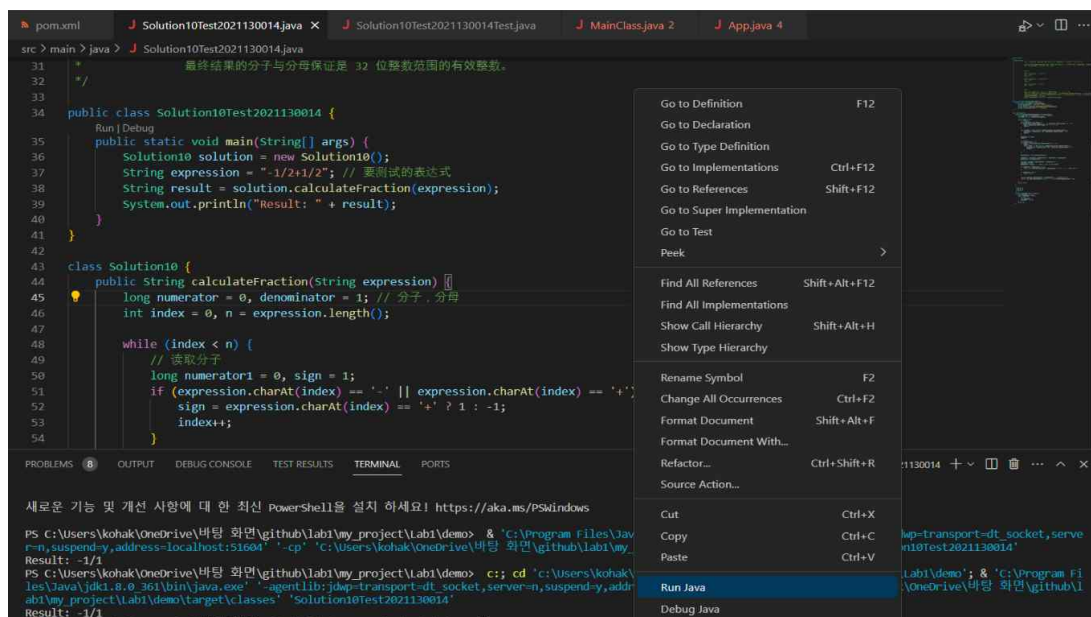
        long g = gcd(Math.abs(numerator), denominator); // 获取最大公约数
        return Long.toString(numerator / g) + "/" + Long.toString(denominator / g);
    }

    return "0/1";
}

/**
 * @param a
 * @param b
 * @return
 */
private long gcd(long a, long b) {
    long remainder = a % b;
    while (remainder != 0) {
        a = b;
        b = remainder;
        remainder = a % b;
    }
}
```

2.4 测试通过截图

将JUnit测试通过后的截图粘贴在此处




```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

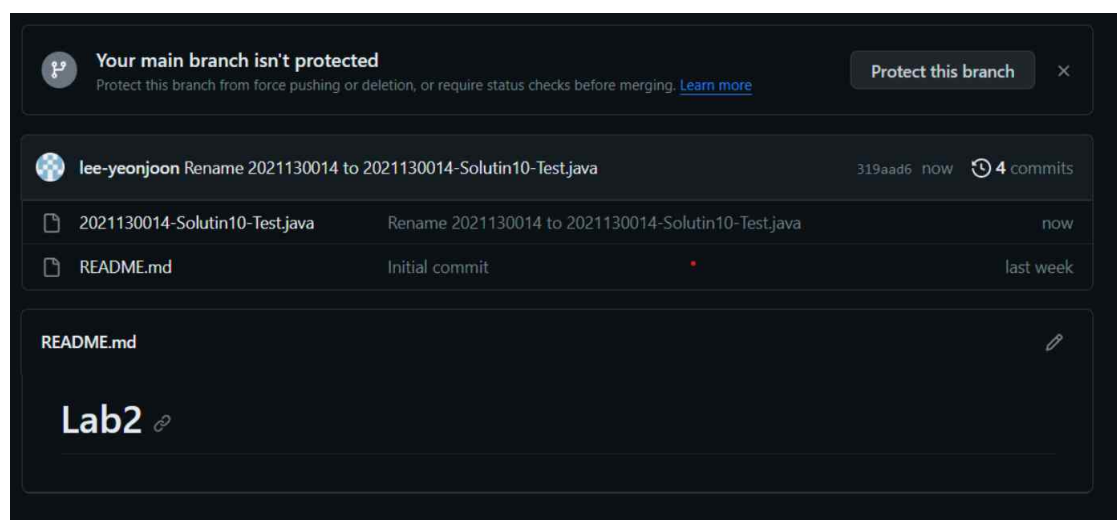
새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 설치 하세요! https://aka.ms/PSWindows

PS C:\Users\kohak\OneDrive\바탕 화면\github\lab1\my_project\Lab1\demo> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' -jar -r=n,suspend=y,address=localhost:51604' '-cp' 'C:\Users\kohak\OneDrive\바탕 화면\github\lab1\my_project\Lab1\demo>
Result: -1/1
PS C:\Users\kohak\OneDrive\바탕 화면\github\lab1\my_project\Lab1\demo>
```

点击 Run Java后，这个代码的结果值是-1/1。

3 实验内容2 接受pull request

将双方通过github上PR交流的记录截图粘贴在此处



push后的GitHub页

4 实验内容3 github辅助工具

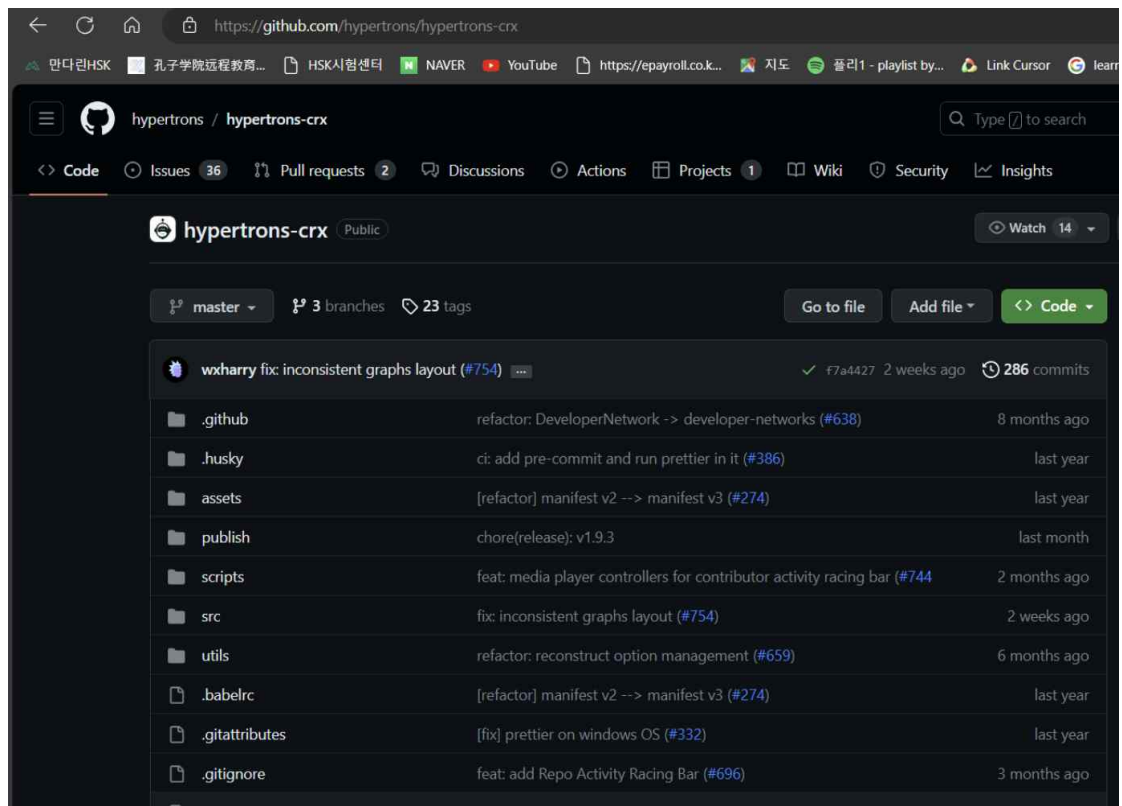
4.1 熟悉GoodFirstIssue工具

熟悉<https://goodfirstissue.dev> 网站的使用

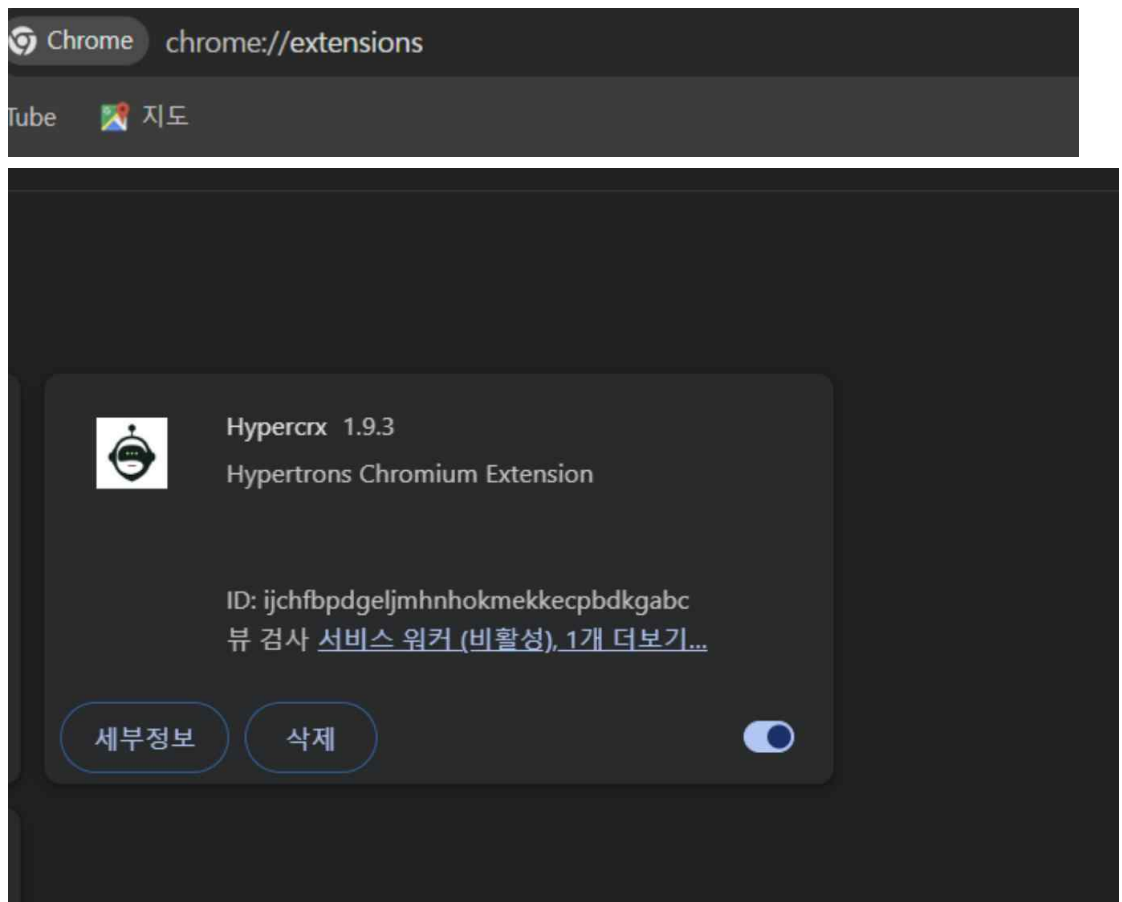
查阅项目的文档，阐述如何使自己的开源项目被此网站收录

4.2 安装并使用Hypercrx

访问此网站<https://github.com/hypertrons/hypertrons-crx>



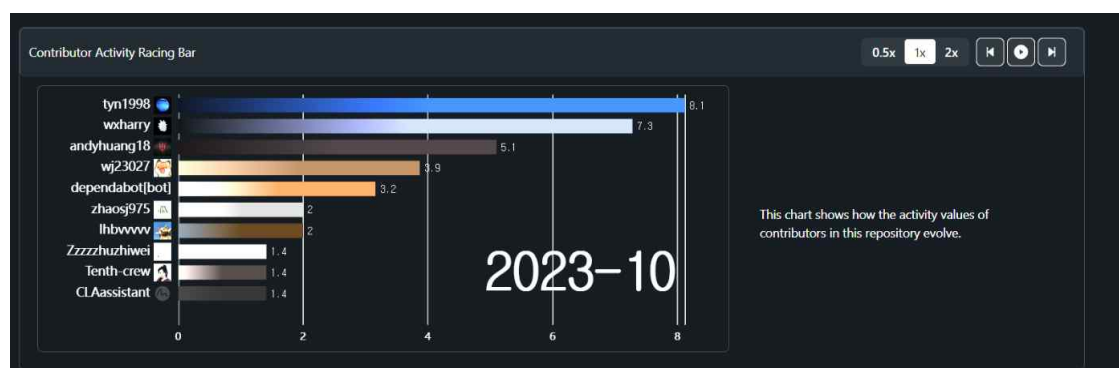
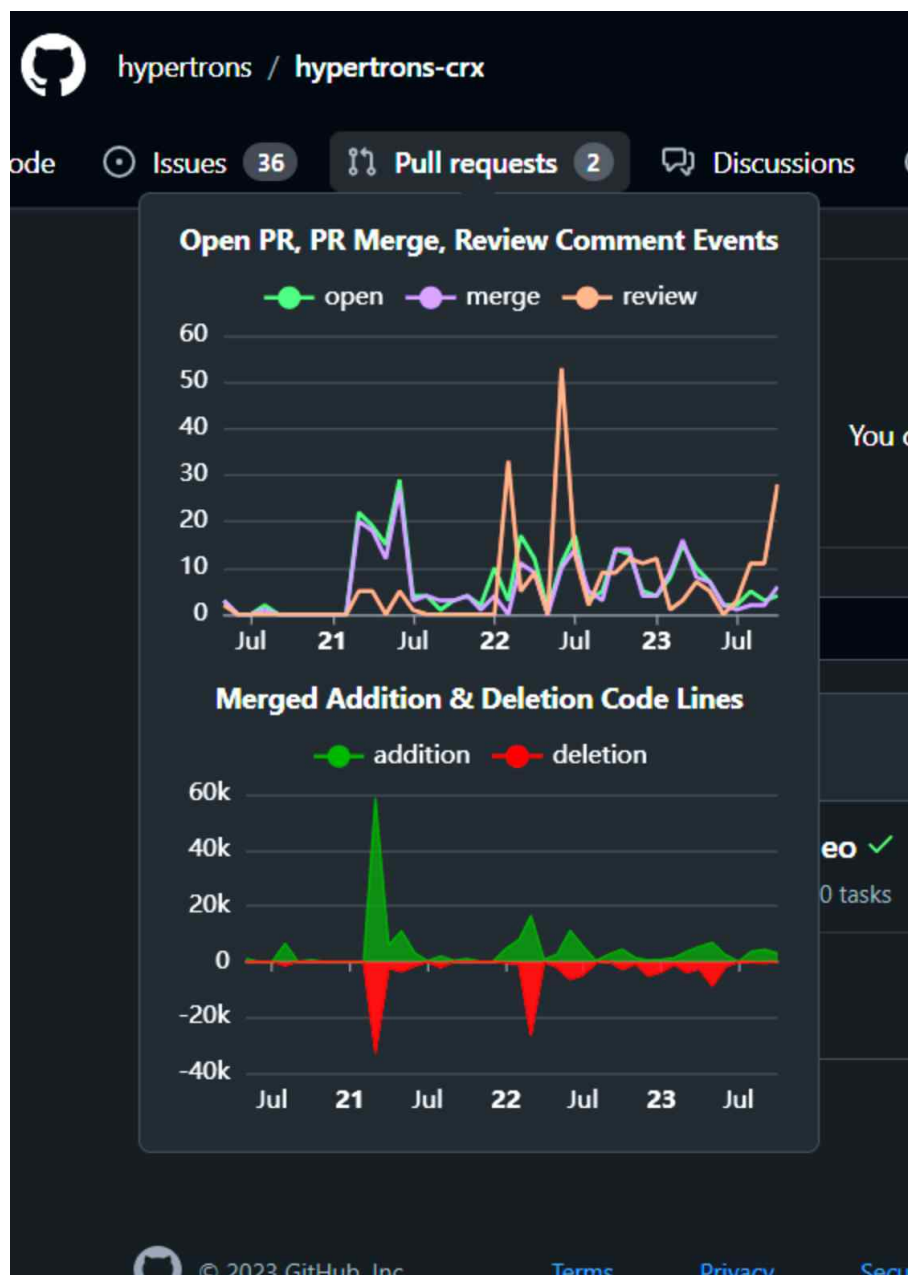
安装Hypercrx插件



打开chrome web浏览器（<chrome://extensions>）后，在chrome webstore下载 Hypercrx。

找到某个star在100+的项目，给出利用插件查询项目状态的截图。例如下图：

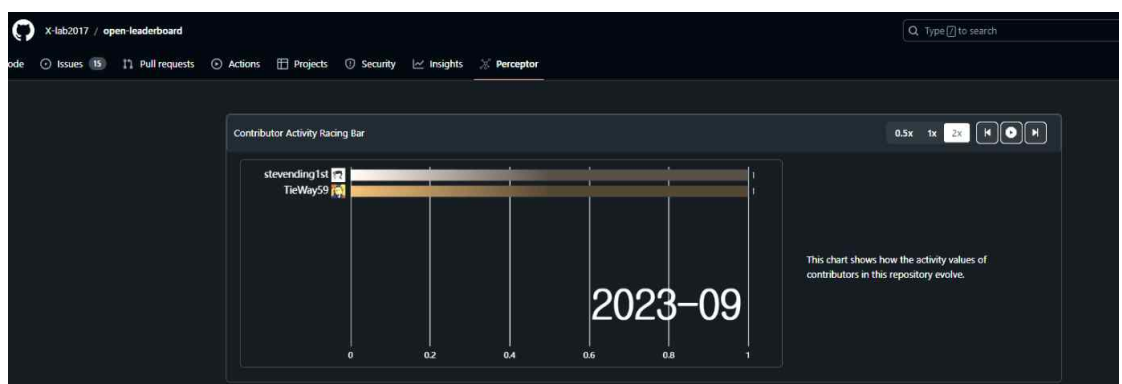
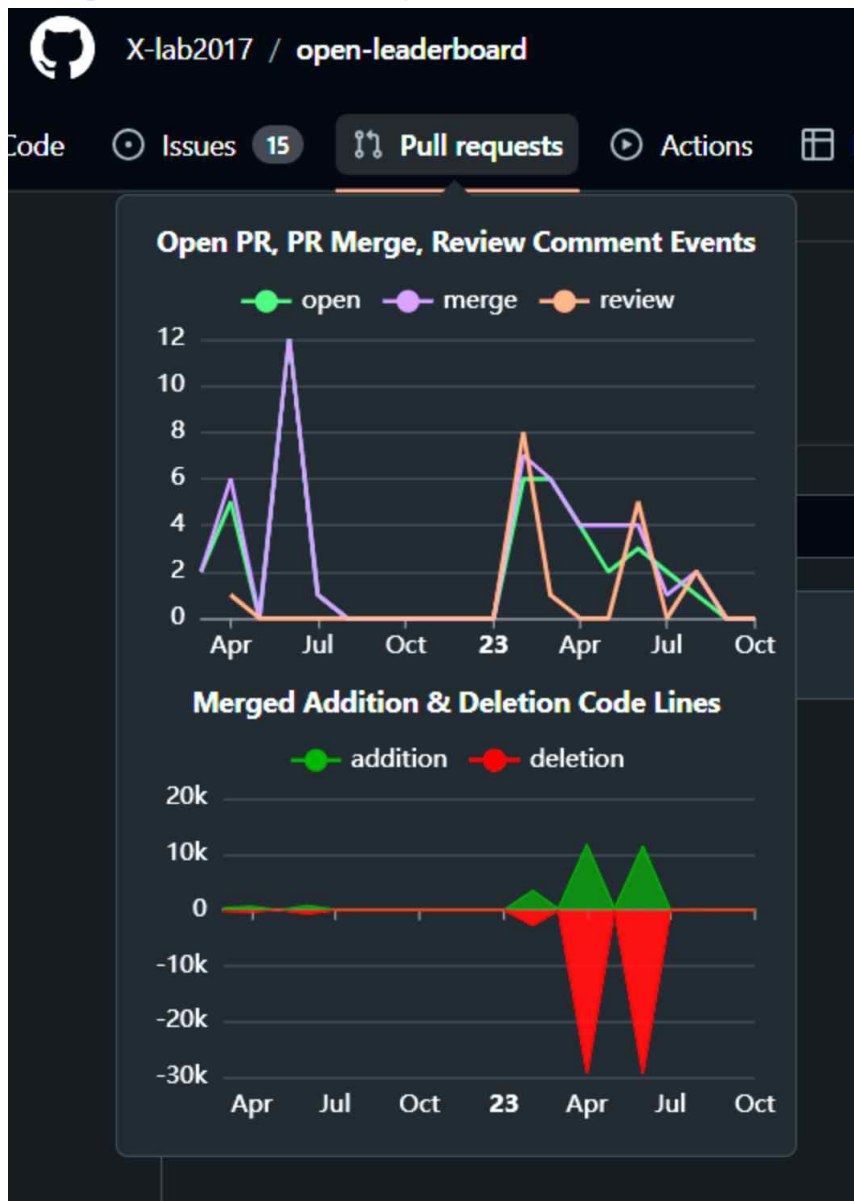
<https://github.com/hypertrons/hypertrons-crx/pulls>



对比下使用此插件前后的效果

4.3 利用OpenLeaderboard工具

利用OpenLeaderboard工具了解github上开源项目的统计情况



阅读实验手册中指定的三篇文章，概括开源项目活跃度、影响力指标、价值流网络的含义，OpenRank的计算原理。

《开源项目活跃度》

作者写作的背景是如何准确评估程序是否健康。他写这篇文章回顾了他迄今为止的工作。

活力-Apache路演于2015年进入开源社区，并参与了许多活动。在那之前，对开源的理解有点模糊，他从未真正参与过。在加入阿里之前，我是根据GitHubstars和大型工厂或基金会的项目进行评估的。这种评估简单而直接，这就是为什么它对许多人来说仍然更引人注目。小狼问题问我有多少项目是在开源，在什么领域，以及如何评估这些项目的好坏。然而，在这一点上，大多数公司对开源社区的量化还没有开始。

在深入研究GitHub数据的基础上，提出了一种基于GitHub行为数据的加权主动算法。该算法的计算公式为 $Ad = \sum w_i c_i$ ，其中Ad表示开发者活动， c_i 是五个动作事件被触发的次数， w_i 是每个动作事件的加权比。该加权比可以具有从1到5的范围内的值。例如，每个问题评论1分，每个问题开始2分，每个PR开始3分，PR的每个代码审查评论4分，PR总计5分。这些值用于计算每个开发人员的活动，然后计算项目活动，此时使用的方程为 $Ar = \sum Ad$ 。这种方法的目的是通过为所有开发人员的活动开出处方来减少对关键开发人员过度活动的影响。

-思考

1.开放源码办公室需要一个“北极星指标”来监控一些项目。总之，北极星指标是对球队目标和方向的衡量，也是对球队目标的衡量。

2.星、叉等行为表示对开源的兴趣，但不包括在活动计算中，因为它们没有反映实际贡献。换句话说，这种行为在活动算法中被认为是无效的。

3.即使基本贡献者被定义为代码贡献者，它也被认为是社区中的所有开发人员对项目的贡献，并且计算不仅包括代码，还包括讨论。

4.这五项活动的权重是主观的，由董事在阿里战略层面上的同意设定。例如，它对PR的审查给予了很高的重视，并建议使用基于GitHub的异步审查。

5.考虑到开发人员活动和项目活动，对每个开发人员的活动进行了解释。这种活动计算方法基于GitHub行为数据，使用专家经验分配和价值导向权重。

这种活动计算方法在GitHub洞察报告中重复使用，并具有特定的价值取向。最近的报告显示，权重调整和采用新指标正在继续发展计算方法。

但计算活动也存在问题。例如：

1.存在一个问题，即如何包括不确定应包括哪些指标且不能提供实质性反馈的行为，特别是在活动概念下。

2.行为的权重是主观的，反映了特定的专家经验，但即使是微小的差异也会对整体活动产生重大影响，这可能导致项目之间的比较困难。

3.活动计算方法缺乏考虑随时间变化的基线，也没有标准，这使得在其他时期进行直观比较变得困难。

4.活动计算方法很难在仓库级别线性添加，这会影响多次操作。

5.个人活动和仓库活动之间的联系是一种经验做法，尽管使用了一般方程来模拟边际利润减少效应，但对数函数也可能是一种有效的替代方法。

6.使用单一指标是不可避免的，而且有一种经验表明，自引入指标体系以来，开发人员的活动发生了变化，我们认为这是该指标可以作为积极指导的证据。

-推广与改进

除了一年一度的《数字洞察报告》，阿里有意不推广这些指标，但在内部，他们继续提

提供服务,以更好地了解开源项目的现状。例如,通过将星叉与活动分开来为项目做出贡献的行为就是活动,而关注项目但没有实质性反馈的行为则被独立地视为兴趣。通过这种方式,可以识别出两种不同的生长模式,也有人试图通过主动分布和总线系统。它还提供了基于协作网络的见解。

在金融开源社区,采用类似指标体系的项目正在增加。例如,PUPHB银行发布了一份文件,将时间序列的一些衰减指数添加到行为数据中。

-结论

尽管活度计算方法目前存在许多问题,但它直观、易于理解,并且具有很强的可解释性,因此目前在实验室中被广泛使用。尽管它已经应用于许多项目,但我们期待着在未来通过更好的指标系统和算法框架来有效地衡量开源生态和网络。

《操作影响力》

-背景

上一集提到了基于活动的统计指标的局限性和问题。这种分析方法可以在某些项目上有效地工作,但由于开源是整个生态系统,GitHub开发人员在多个项目上工作,因此它没有考虑项目之间的关系,并且没有充分利用自动化行为或GitHub功能的项目可能会出现问題。

对于具有大量自动化行为的项目或未完全使用GitHub功能的项目,全局活动分析可能会导致问题。考虑到这一点,通过提出一个开源合作网络,我们设定了一个目标,即通过项目之间的合作来计算对开源生态系统中所有项目的影响。

-开源协作网络与协作影响力

1) 开源写作网络

开源协作网络的想法很简单,假设如果开发人员同时参与两个项目,那么这两个项目之间存在高度的协作相关性。研究表明,这种关系在大多数情况下源于项目之间的上下游关系或使用依赖和合作关系。例如,在VSCode和Flutter之间的连接中,协作机器人的普遍使用和开发人员Danny Tuppeny的高度活跃将这两个项目联系在一起。发现这些关系的一个有用的数学工具是协作网络中的协作相关性计算,用于根据开发人员的活动评估关联。我们使用调和平均计算方法来评估开发人员的贡献,只有当开发人员同时在两个项目中非常活跃时,这种方法才会对两个项目的关联产生更大的影响。

开发者d在项目p1中具有活动Ad, p1, 在p2中的活动为Ad, p2。换句话说,该开发者将两个程序的协作相关性的贡献表示为 $Adp1 * Adp2 / (Adp1 + Adp2)$ 。

2) 协作影响力

我们使用PageRank算法在开源的本地生态系统中建立了一个合作网络,并以此为基础计算合作影响。该算法被谷歌搜索引擎使用,是根据网页之间的引用关系确定页面排名的有效算法。

PageRank通过页面之间的引用关系来确定页面的影响,假设高质量的页面被更多的页面引用,并且被引用页面的质量也更高。将这一想法应用于开源协作网络,协作项目越多,项目的影响力就越大,而影响力越大的项目,协作项目的影响力也就越大。

有了这一认可, GitHub在2019年的全域协作项目包括:

排名	项目	写作影响力
1	microsoft/vscode	1135
2	flutter/flutter	645
3	kubernetes/kubernetes	624
4	DefinitelyTyped/DefinitelyTyped	564
5	microsoft/TypeScript	544
6	tensorflow/tensorflow	535
7	gatsbyjs/gatsby	504
8	golang/go	448
9	rust-lang/rust	448
10	facebook/react-native	426

-思考

- 1.协作影响是基于活动的，但它利用来自开源生态系统的全球数据来避免与活动相关的问题，并包含重要的相关信息。
- 2.自动化项目中的高活动性被从网络中删除，因为与其他无关项目没有合作关系。
- 3.对于刷动，如果另一个生态项目的开发商不活跃，那么高的活动分数不会显著影响项目的影晌。
- 4.不使用GitHub Issue功能的项目在其他上下项目中都是活跃的，因此比活跃的指标具有更好的影响力。
- 5.由于使用了网络关系信息，活动计算中的权重变化对影响力排名的影响较小。
- 6.构建协作网络提供了额外的分析能力，以及协作影响计算，并可以评估协作高程等方面。

但协作影响力的问题是：

- 1.开源协作网络只分析当前项目，在分析开发人员时使用类似的网络限制了自动协作帐户的好处。
- 2.该模型仅使用开发人员的GitHub行为数据，并且存在无法获得大量数据的限制。
- 3.PageRank算法的初始值不相关是模型中没有考虑项目或开发人员的唯一属性的缺点之一。
- 4.开源协作网络的设计对活动不敏感，但在活动中引入低成本行为会导致多个项目之间的连接，从而降低项目类别确定的准确性。集群的有效性在很大程度上取决于当前集群的活动设计。

-总结

基于开源协作网络的项目协作影响力是解决基于统计的活动指标问题的有效手段，也是有效评估和深入了解整个开源生态系统的项目影响的有效手段。然而，整体模型和指标的可扩展性仍然存在限制，我们希望找到利用更多开源数据的方法，并开发具有高可扩展性的图模型，以提供更深入、更准确的见解和分析。此外，我们希望避免指数得分带来的一些问题。

《价值流网络》

-背景

在以前的博客中，我们已经介绍了如何计算活动指数以及如何计算项目协作影响指数。考虑到各种行为权重和协作关联，这些方法在衡量开源生态系统中开发人员和项目的

影响力方面是有效的。然而，这种方法有几个问题和局限性，并且有显著的缺点，需要在引入新数据时修改模型。这需要一个高度可扩展的数学模型，可以整合更多的开放数据。因此，我们将在本章中介绍价值流网络。

-价值流网络

价值流网络考虑了衡量开源软件价值的两个重要方面：生产和消费。它直接衡量每个软件的社会价值，考虑到开发人员之间的差异以及项目是否被使用，同时创建一个可以识别开发人员价值的模型。这使您能够准确地理解开源软件所创造的社会价值，并公平地评估开发人员的成就。这一分析对于形成一个完整的开源经济生态系统具有重要作用。

-从协作影响力说起

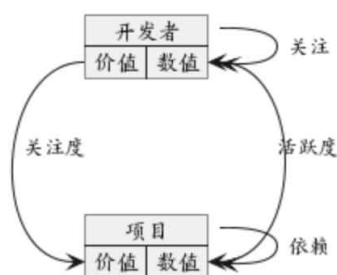
网页排名算法中的概率模型基于用户的网页移动模式来确定页面的值。每个网页通过外部链接传递其价值，并从链接的网页中获取价值。一旦整个网络结构中的价值流稳定，就会确定每个网页的价值，并根据其排名靠前。

同样，在开源协作网络中，每个项目都有一个基本价值，价值的转移和流动是由开发人员之间的协作产生的项目之间的关联产生的。所有项目的协同影响都是确定的，直到整个网络稳定为止，这可以衡量每个项目的相对价值。

一个简单的示例

从以上角度来看，它实际上可以提供一种更通用的方法来构建和计算价值网络。下面显示了添加更多数据关系的最简单示例，特别是在生产和消费方面，同时考虑开发人员的贡献活动。

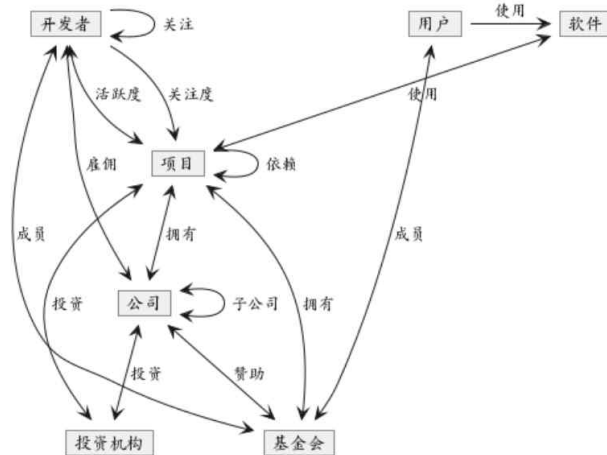
上面的价值流网络代表了开发者和项目之间的互动，每个节点都有自己的价值。开发人员通过对项目的活动和兴趣来传达价值，而项目则基于依赖和活动来传达价值。此外，开发商对该项目的兴趣也会作为一个特别的提醒出现。



在这个模型中，全球开源项目和开发人员的价值是通过活动、项目兴趣和对其他开发人员的兴趣来传达的。开发人员创造的一些价值被泄露给了通过他们的行动做出贡献的项目，还有一些转移到了表达兴趣但没有真正做出贡献的工程中。项目的价值通过活动和依赖返回给开发人员，其中一些会转移到父项目。这样，个人创造的价值是由他们的活动和项目之间的关系驱动的，整个网络中的价值流决定了每个节点的相对价值。

-开源生态价值流网络

虽然简化形式的数学模型易于验证且具有高度可扩展性,但开源生态系统的价值网络实际上要复杂得多。从生产到消费的数据各不相同,尤其是在消费方面,依赖其他项目并不是开源项目最终消费的方式。所有软件最终都会变成一种服务,并以满足真实用户需求的方式进行消费,如果项目没有被任何用户使用,没有满足现实需求,或者创造社会效用,那么最终价值就会受到限制。



在上述网络结构中,价值流考虑了各种实体之间的关系,包括项目、软件、开发商、公司、基金会、投资机构和用户。价值通过多种渠道传递,包括开发者活动、用户软件使用、投资机构投资、雇佣和赞助公司以及基金会赞助,每个实体之间的关系代表了网络内的价值流。

该模型有助于评估开源软件的生态价值,旨在通过量化每个部分的具体价值和流动机制来评估实际的社会效用和价值。最后,随着数据的不断完善,这个值预计将更接近实际的社会效用。

这种分析是用于理解和建模系统复杂性的常用方法之一,有助于识别价值流和实体之间的互动。当然,还必须考虑安全风险等其他方面,但该模型为开源生态系统的价值和互动提供了一个重要的视角。

-思考

- 1.价值流网络模型试图提供一个顶级模型,该模型包含开源生态系统中实际存在的尽可能多的数据,并与更多数据兼容。
- 2.该模型将数学模型与商业模型分离,允许上层开源生态描述几乎独立于下层数学模型进行。例如,更复杂的网络结构不包括数学模型。
- 3.模型能否获得稳态解与楼层数学约束密切相关,相关信息可参考附录部分。然而,一旦确定了商业模式,具有基础知识的学生就可以协调和收敛。
- 4.该模型最终试图解决构建整个开源生态系统的经济系统的问题。

-问题

- 1.该模型将数学模型与商业模型分离,具有高度的可扩展性,但除非设计商业模型的人熟悉底层数学逻辑,否则很难获得可靠的解决方案。因此,对商业模型设计者的要求可能会提高。
- 2.模型需要大量数据来准确衡量整个开源数字生态系统的经济系统,其中许多可能很难获得,也很难找到关联。这需要长期和大规模的合作,并将是未来这方面工作的重要组成部分。

5 小结

对本次实验过程和结果的思考、对本次实验的收获、对本次实验的建议等。

修改代码是实验中最困难的部分。有一个错误，所以如果你纠正了那个部分，另一个部分也发生了错误，即使你纠正了发生的错误，也发生了另一个错误。你似乎陷入了错误纠正的无限束缚。我想过中途放弃，但当我纠正了所有的错误并通过了测试并获得了结果值时，我有了很大的成就感和信心。

我认为这个实验最大的收获是知道如何通过hypercrx检查GitHub项目的状态，以及通过代码纠错来保持持久性和耐心。通过一个名为hypercrx的新扩展程序，当他们打开PR、合并PR和审查PR时，了解内容的历史是一种新的体验。

我认为可以使用更多样化的工具。