用户故事编号	用户故事描述（作为XX，我希望做XX，以便XX）
3	作为用户，我希望能够修改自己的名字，以便展现个性
4	作为用户，我希望能够创建群，以便能够和更多的用户同时交流
5	作为用户，我希望能够通过输入关键词来查询自己加入的所有群的名称，得到所有包含该关键词的所有消息，以便快速查找历史消息。
	（如果愿意，可以在该行内填写更多的用户故事）

软件测试（30分）

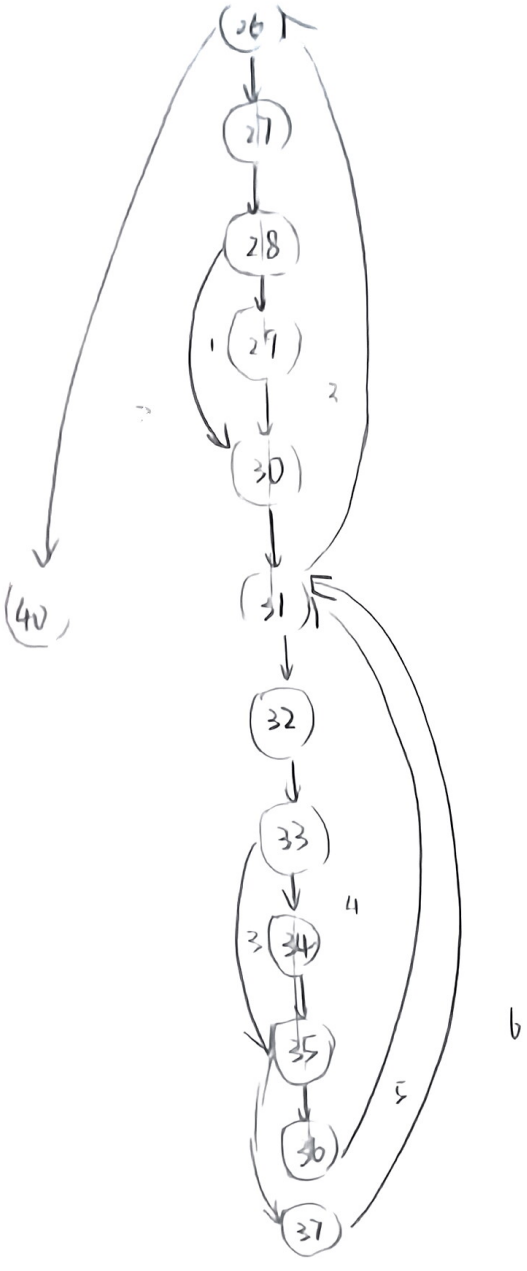
假设微信的源代码中包含如下代码片段：

```
1   public class SocialNetwork {

2       private ArrayList<String> nodes = new ArrayList<String>();
3       //such as {"a","b","c","d"} 表示微信社交网络中的用户集合
4       private ArrayList<String> edges = new ArrayList<String>();
5       //such as {"a-b", "a-d", "c-d"} 表示社交网络中社交关系的集合

        //检查用户uID1和uID2之间是否存在社交关系
6       public boolean checkExist(String uID1, String uID2) {
7           if (edges.contains(uID1 + "-" + uID2)
                || edges.contains(uID2 + "-" + uID1)) return true;
8           return false;
9       }
        //在用户uID和用户friendID之间添加一条代表社交关系的边
10      public void addRelation(String uID, String friendID) {
11          edges.add(uID + "-" + friendID);
12      }
        //向社交网络中增加一个用户
13      public void addUser(String id) {
14          nodes.add(id);
15      }
        //检查社交网络中是否已包含某个用户
16      public boolean findUser(String id) {
17          if (nodes.contains(id)) return true;
18          return false;
19      }
```

1) （5 分）控制流图



① 26-40
② 26-27-28-30-31-32
③ 26-27-28-29-30
④ 26-27-28-30-31-32-33
⑤ 26-27-28-30

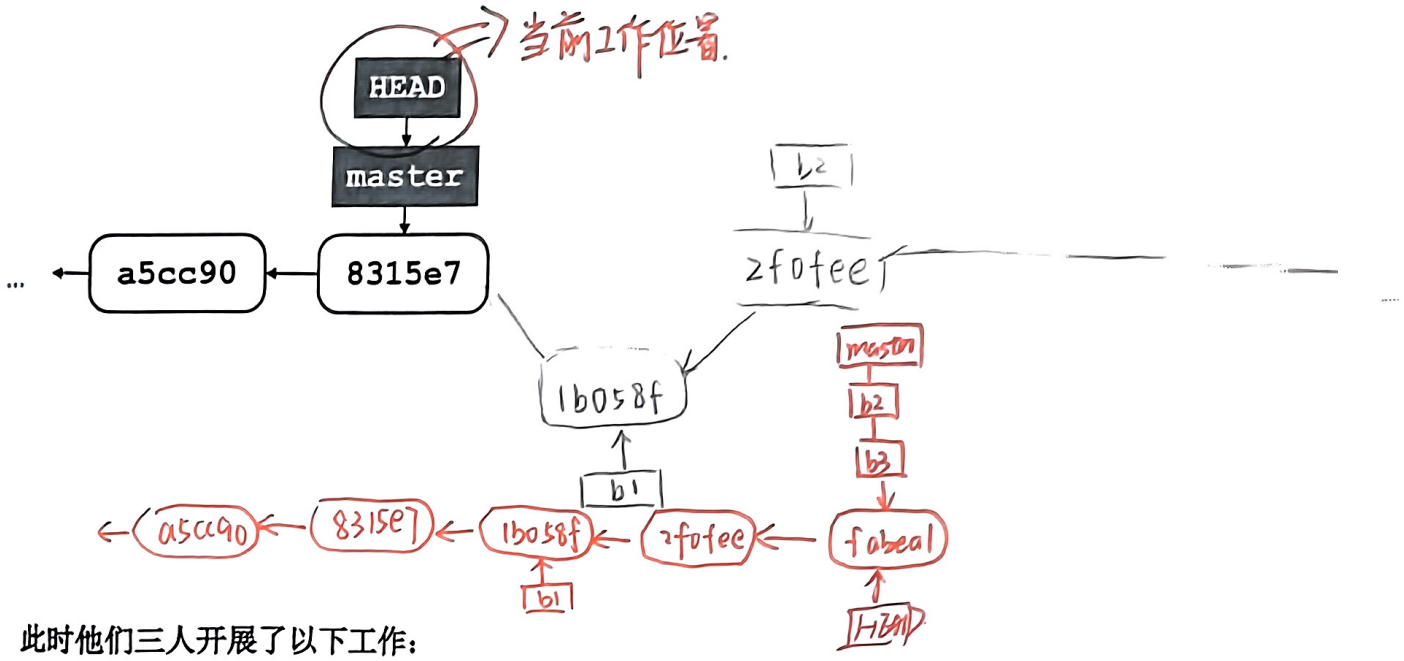
根据用例写路径编号.

2) （9 分）基本路径（按需填写，未必要全部填满）

路径编号	路径的具体构成（使用代码行号表示，例如 1-2-3-4-5）
1	① 26-40　④ 26-27-28-29-30-31-32-33-34-35-37-31-26-27-30-31-32-35-36-31-26-40
2	~~26→27→28→30→31→40~~
3	① 26→27→28→29→30→31→40
4	~~26→27-28-30-31-32-33-35-36-31-40~~
5	~~26-27-28-30-31-32-33-34-35-36-31-40~~
6	③ 26-27-28-30-31-32-33-34-35-37-31-40
7	

配置管理（10 分，每题 5 分）

有三个开发者参与一个项目，A 负责开发初始代码，B 负责修复 bug 和优化代码，C 负责测试并报告 bug。项目的 Git 服务器为 S，三人的本地 Git 仓库已经配置好远程服务器（名字均为 origin）。项目的 Git 版本状态如图所示，三人的本地 Git 仓库的状态也是如此，其中包含主分支 master，当前工作分支是 master。



此时他们三人开展了以下工作：

a) A 开发了某项新功能，他创建了分支 b1 并在该分支上提交新代码，推送至服务器 S；
b) C 获取了 A 的提交，并在其上开辟了新分支 b2，在 b2 上撰写测试程序并提交和推送至服务器 S；
c) C 在执行测试程序过程中发现了 A 的代码存在问题，C 将 bug 信息报告给 B；
d) B 获取了 C 推送的包含测试程序的版本，在其基础上开辟了一个新分支 b3 用于 bug 修复，当 B 确认修改后的代码可通过所有测试用例之后，向 Git 做了一次提交，将 b3 合并到 b2 上并推送至服务器 S；
e) C 获取 B 的修复代码并重新执行其中包含的测试程序，确认 bug 已被修复，故将其合并到主分支 master 上，推送至服务器 S，对外发布。⇒没有修改提交.

题目：

(1) 在图上补全上述活动结束后服务器 S 上的版本状态图（需注明各分支的名字与位置）；
(2) 写出 B 为完成步骤 d 所需的全部 Git 指令，指令需包含完整的参数。

git fetch / pull origin b2

git checkout -b b3 origin/b2 基于这个分支状态创建.

git commit -am "bug 修复"

git checkout b2

get merge b3

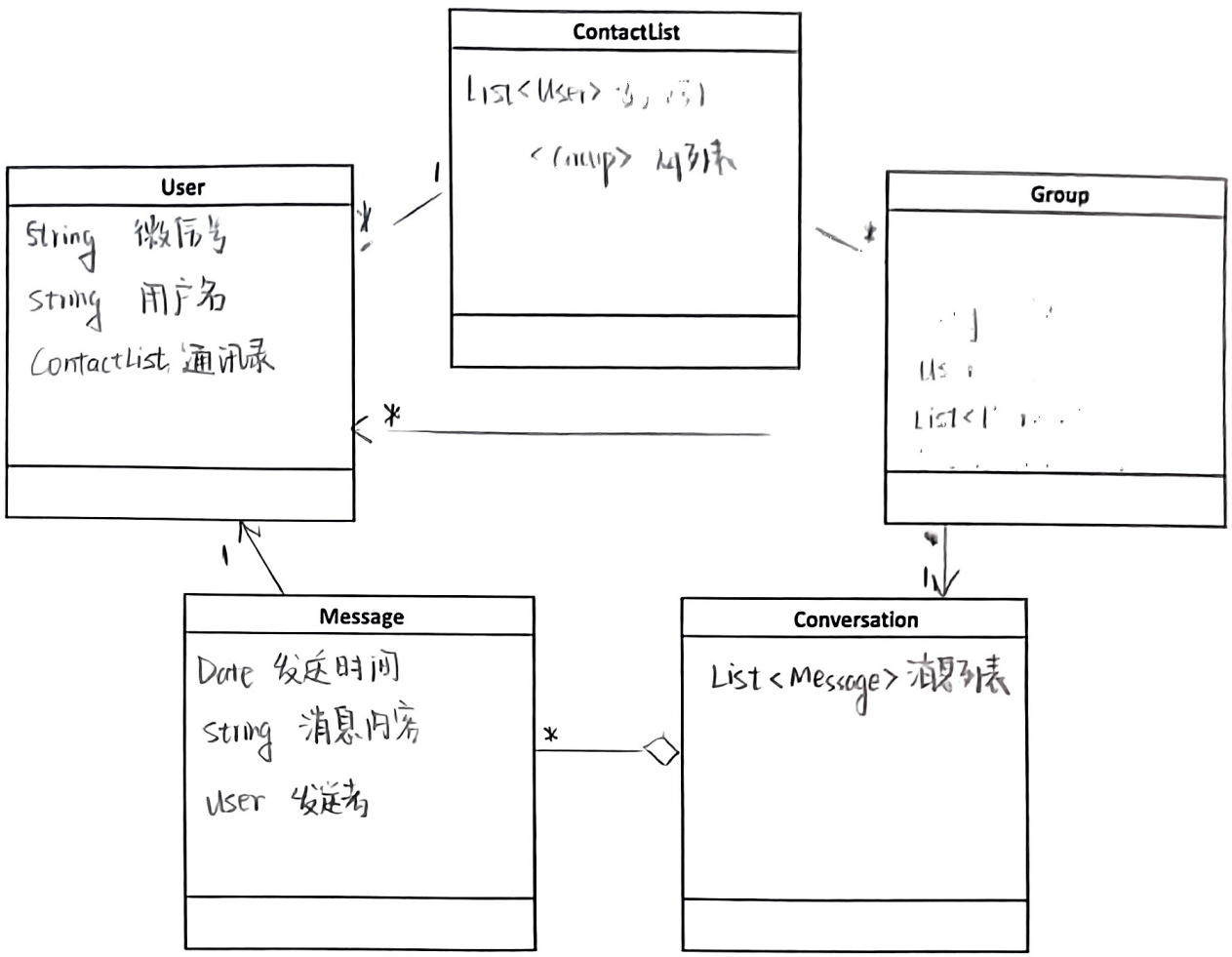
get push origin b2

授课教师 姓名 学号 院系

密 封 线

五 面向对象分析（25 分）

(1) （9 分）第 1 页的需求描述中提及了以下实体类：用户（User）、用户的通讯录（ContactList）、群（Group）、消息（Message）、群会话（Conversation）。在下图中的相应区域填入实体类的属性（格式：数据类型 属性名，每行一个）。请务必包含各关联属性。属性名请使用中文。



(2) （8 分）识别这些实体类之间的静态结构关系，在上图中补充绘制这些关系（至少包含：线或箭头、关系的多重性）使该图成为完整的领域类图。由于空间狭小，请确保补充后仍清晰可读。

组合、聚合 / 关联？

(3) （8 分）“退群”这个动作最适合作为哪个实体类的操作？思考一下该操作的内部实现逻辑，它本质上对该实体类的哪个（些）属性做了何种更新？除了该实体类，还会改变其他哪个（些）实体类的什么属性？具体做了何种更新？

① 适合作为 Group 操作

② 对 Group 类的成员列表中移出该用户

③ 还会改变该用户的通讯录 从好友列表中移出该好

ContactList 、

自己去除好

六 简答题（15 分，任选两题作答，若回答全部问题，按前两题评分）

(1) （9 分）本学期 Lab1 要求开发一个 Java 程序，从文本文件中读取数据并生成有向图，进而在图上进行若干操作，其中有一个操作是“随机游走”，其函数规约为 `String randomWalk(type G)`。如果要对该函数进行黑盒测试，设计测试用例并用 JUnit 编写测试程序，但由于该函数的内部实现需要多次执行随机函数，这导致其输出结果不固定，多次执行所产生的路径可能均不同，故 JUnit 测试函数中无法使用 `assertEquals()` 来判定程序是否通过测试。如何解决？

① 固定性质而非固定输出：可以测试 { 有效性 — 返回路径在图 G 中有效；完整性 — 图中如果存在无出度的顶点，则路径应停在到达该停止；长度 — 根据图的结构，测试返回长度是否符合预期 }

② 控制随机性：固定随机种子

(2) （6 分）动态代码评审工具采集被测程序运行时的数据，进而评估程序的时空性能。目前的动态代码评审工具有两种实现策略：采样、代码注入。请结合你在本学期 Lab4 中使用 VisualVM 进行 Java 程序动态评审的体验，简要阐述二者在分析代码性能的能力与性能上有何差异。

4-1 P7 采样 注入.

采样：{ 低开销 快速有效；精度低、不完整 }

代码注入：{ 高精度、详细数据；高开销、可能影响程序 }

(3) （6 分）“缓存（Cache）”和“分布式集群”均可用来改善 Web 系统在大规模并发情况下的系统响应时间（即用户发出请求到得到系统反馈所需要的时间）。请简要分析这两种架构设计策略分别以什么原理来改善“响应时间”？有什么差异？

见补充资料 + 5.2 P80.