一、正常运行 jar 包的操作

1、 运行 jar 包，首先会让你选择是生成索引还是读入索引，生成索引输入 1，读入索引输入 2

2、 如果选择生成索引，接下来会分别让你输入读入的文件路径和索引输出的文件夹路径，在输入两者后，在文件夹路径会生成多个与图有关的文件，分别记录 index 和 truss 等信息，之后程序会不断循环的让你输入 node id 和 k 值，然后输出结果直到你输入-1 选择终止。

```
You want to calculate index(1) or search community(2)
1
Please input the filename
data/toyg.txt
# of vertices: 34
# of edges: 78
# dmax: 17
graph read time: 0.003795195
Please input the index path
toy
creating directory: toy
support time:0.001997978
computeSupport complete!
sort complete!
#kmax : 5
truss computation time: 0.005596556
Index for given graph is created. Index creation time: 0.00188
enter query node id and k:truss value
4
4
Number of edges in this community: 14
(3,8), (1,8), (3,1), (2,8), (2,3), (4,8), (4,3), (2,1), (4,1),
2,4), (2,14), (14,4),
query time:1.9803E-4


enter query node id and  k truss value, -1 to exit
-1
```

3、 如果选择读入索引，也需要输入文件路径和索引的文件夹路径，在程序读取图和 index 信息后，同上，会不断循环的让你输入 node id 和 k 值，然后输出结果直到你输入-1 选择终止。

二、复现实验

统计 F1-score

实验中唯一需要加的代码就是统计 F1-Score，要运行改源代码中 main 包里

的 Main class 的 main 方法，需要把 main.test(tec);改成 main.testefficiency(tec);，在源代码里面已经有了 facebook 的图数据和社交圈信息，所以只需要在运行时照常选择 1，输入文件和文件夹路径，之后就会输出 10 个点不同 k 下的 F1-Score 以及平均的 F1-Score。

计算任意给定的 100 个点的总搜索时间

实验中需要对运行效率进行评估，在两种评估角度中都需要计算给定 100 个点进行社区搜索的时间。我们将这么一个任务抽象为源码中 experiment 包里的 Main class 的 main 函数，对原始图文件路径、索引等存放的文件夹路径、truss k、保存待搜索的 100 个点的 index 的文件路径这四个变量进行设置之后运行 main 方法即可进行计算。