```
1 "C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" "-
  javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
  Community Edition 2020.2.1\lib\idea_rt.jar=57853:C:\
  Program Files\JetBrains\IntelliJ IDEA Community
  Edition 2020.2.1\bin" -Dfile.encoding=UTF-8 -
  classpath "C:\Program Files\Java\jdk1.8.0_261\jre\lib
  \charsets.jar;C:\Program Files\Java\jdk1.8.0_261\jre\
  lib\deploy.jar;C:\Program Files\Java\jdk1.8.0_261\jre
  \lib\ext\access-bridge-64.jar;C:\Program Files\Java\
  jdk1.8.0_261\jre\lib\ext\cldrdata.jar;C:\Program
  Files\Java\jdk1.8.0_261\jre\lib\ext\dnsns.jar;C:\
  Program Files\Java\jdk1.8.0_261\jre\lib\ext\jaccess.
  jar;C:\Program Files\Java\jdk1.8.0_261\jre\lib\ext\
  jfxrt.jar;C:\Program Files\Java\jdk1.8.0_261\jre\lib\
  ext\localedata.jar;C:\Program Files\Java\jdk1.8.0_261
  \jre\lib\ext\nashorn.jar;C:\Program Files\Java\jdk1.8
  .0_261\jre\lib\ext\sunec.jar;C:\Program Files\Java\
  jdk1.8.0_261\jre\lib\ext\sunjce_provider.jar;C:\
  Program Files\Java\jdk1.8.0_261\jre\lib\ext\sunmscapi
  .jar;C:\Program Files\Java\jdk1.8.0_261\jre\lib\ext\
  sunpkcs11.jar;C:\Program Files\Java\jdk1.8.0_261\jre\
  lib\ext\zipfs.jar;C:\Program Files\Java\jdk1.8.0_261\
  jre\lib\javaws.jar;C:\Program Files\Java\jdk1.8.0_261
  \jre\lib\jce.jar;C:\Program Files\Java\jdk1.8.0_261\
  jre\lib\jfr.jar;C:\Program Files\Java\jdk1.8.0_261\
  jre\lib\jfxswt.jar;C:\Program Files\Java\jdk1.8.0_261
  \jre\lib\jsse.jar;C:\Program Files\Java\jdk1.8.0_261\
  jre\lib\management-agent.jar;C:\Program Files\Java\
  jdk1.8.0_261\jre\lib\plugin.jar;C:\Program Files\Java
  \jdk1.8.0_261\jre\lib\resources.jar;C:\Program Files\
  Java\jdk1.8.0_261\jre\lib\rt.jar;C:\Users\Matth\
  Desktop\Assignment_4\Assignment_4\out\production\
  Assignment_4" WordNet
2 edges: 18419
3 Amount of Modules: 1200
4 Top 20 Modules: 1338 46 32 23 15 14 9 8 8 8 8 7 7 7 7
   6 6 6 5 5
5 Breadth-First Search Algorithm: money -> pay -> raise
   -> bring -> future
6 Dijkstra's Search Algorithm: money -> pay -> raise
   -> bring -> future
7 Same!
8
9 Breadth-First Search Algorithm: village -> town ->
```

```
 9 city
10 Dijkstra's Search Algorithm: village -> town -> city
11 Same!
12
13 Breadth-First Search Algorithm: bad -> too -> good
14 Dijkstra's Search Algorithm: bad -> really -> good
15 Different!
16
17 Breadth-First Search Algorithm: problem -> because
    -> make -> opportunity
18 Dijkstra's Search Algorithm: problem -> because ->
   come -> opportunity
19 Different!
20
21 MST Cost: 2958.620713525721
22 Execution Time (s): 25
23
24 Process finished with exit code 0
25
```