Part A:

- BruteGenerator

    i. the length of the training text

    hypothesis: I think that the run time will increase in a linear fashion, so the big O notation will be O(N). In generateText we loop over the file up through the file length-1 times and increment by one each time. I don’t square it, triple it, etc. The while loop inside the for loop checks a condition, so it does not alter the Big O notation.

    data: caused an overall, roughly constant increase in the mean runtime (When the length of the text doubled, the runtime doubled, etc.). This supports my hypothesis that the relationship would be linear.

-

    ii. the k-value or length of the word

    hypothesis: When we write Brutegenerator, k is one of the parameters. It defines the size of the NGram. We do not alter/update the value of k in train or in generateText, therefore the runtime will be constant regardless of the value of k. The Big O time should be O(1).

    data: increasing the value of k does not affect the runtime in any noticeable way. There is no pattern to be seen (increasing or decrasing)

-

    iii. the length of the filen text

    hypothesis: The while loop is iterating through the text file and changing the start position each time, increasing it by 1. The while loop is also nested in a for loop, so I think that the Big O time will be O(N^2).

data: increasing the random text length causes a recognizable, consistent increase in the mean runtime

- data:
- Varying k, using random text length 100 and file length 152145 (alice.txt)
- k: 1   mean: 2.931518   stddev 0.008092   ci: [2.915658, 2.947378]
- k: 2   mean: 2.989485   stddev 0.050152   ci: [2.891188, 3.087782]
- k: 3   mean: 2.989316   stddev 0.022296   ci: [2.945616, 3.033015]
- k: 4   mean: 2.974081   stddev 0.009555   ci: [2.955354, 2.992808]
- k: 5   mean: 3.036229   stddev 0.058113   ci: [2.922328, 3.150130]
- k: 6   mean: 2.946243   stddev 0.024688   ci: [2.897855, 2.994631]
- k: 7   mean: 2.864352   stddev 0.040510   ci: [2.784952, 2.943751]
- k: 8   mean: 2.913884   stddev 0.070455   ci: [2.775792, 3.051975]
- k: 9   mean: 2.945434   stddev 0.058546   ci: [2.830684, 3.060185]
- k: 10 mean: 2.950934   stddev 0.065146   ci: [2.823248, 3.078621]
- k: 11 mean: 2.975048   stddev 0.119693   ci: [2.740449, 3.209647]
- k: 12 mean: 3.164657   stddev 0.267129   ci: [2.641084, 3.688231]
- k: 13 mean: 3.117362   stddev 0.211095   ci: [2.703616, 3.531107]
- k: 14 mean: 3.142113   stddev 0.381205   ci: [2.394950, 3.889275]
- k: 15 mean: 3.195019   stddev 0.257741   ci: [2.689846, 3.700192]
-
- Varying text length, using k 5 and file length 152145 (alice.txt)
- text length: 20   mean: 0.600584   stddev: 0.002667   ci: [0.595356, 0.605812]
- text length: 40   mean: 1.187773   stddev: 0.005560   ci: [1.176876, 1.198670]
- text length: 60   mean: 1.762373   stddev: 0.005581   ci: [1.751434, 1.773312]
- text length: 80   mean: 2.361073   stddev: 0.004598   ci: [2.352061, 2.370084]
- text length: 100   mean: 2.980802   stddev: 0.058390   ci: [2.866357, 3.095247]
- text length: 120   mean: 3.551841   stddev: 0.028685   ci: [3.495619, 3.608063]
- text length: 140   mean: 4.149184   stddev: 0.016281   ci: [4.117273, 4.181095]
- text length: 160   mean: 4.749432   stddev: 0.050688   ci: [4.650083, 4.848781]
- text length: 180   mean: 5.354581   stddev: 0.024143   ci: [5.307262, 5.401900]
- text length: 200   mean: 6.009928   stddev: 0.069140   ci: [5.874413, 6.145442]
- text length: 220   mean: 6.508391   stddev: 0.139657   ci: [6.234662, 6.782119]
- text length: 240   mean: 7.025228   stddev: 0.049650   ci: [6.927914, 7.122542]
- text length: 260   mean: 38.666341   stddev: 27908.079672   ci: [-54661.169816, 54738.502499]
- text length: 280   mean: 91.018114   stddev: 57938.104630   ci: [-113467.666960, 113649.703188]

```
• text length: 300    mean: 8.901149     stddev: 0.099145   ci: [8.706825, 9.095472]
•
• Varying file length, using k 5 and text length 100
• unique keys: 4439   mean: 0.074285     stddev 0.000102    ci: [0.074085, 0.074486]
• unique keys: 4823   mean: 0.082433     stddev 0.000084    ci: [0.082268, 0.082599]
• unique keys: 5953   mean: 0.101722     stddev 0.000127    ci: [0.101472, 0.101971]
• unique keys: 12946  mean: 0.225467     stddev 0.000473    ci: [0.224539, 0.226394]
• unique keys: 13095  mean: 0.235207     stddev 0.000393    ci: [0.234438, 0.235977]
• unique keys: 82131  mean: 1.535359     stddev 0.003797    ci: [1.527917, 1.542800]
• unique keys: 152141       mean: 3.031022     stddev 0.038159    ci: [2.956229, 3.105814]
• unique keys: 153080       mean: 3.017983     stddev 0.042070    ci: [2.935526, 3.100439]
• unique keys: 496756       mean: 10.144067    stddev 0.238190    ci: [9.677214, 10.610920]
```

- Map Generator
    i. the length of the training text

Hypothsis: I think the run time will be O(N) since there is one for loop in generateText.

- data: The data supports this because as the text length doubles, the run time doubles as well. As the text length triples, the run time triples, etc. The relationship is linear.

    ii. the k-value or length of the word

hypothesis: The k value is an input value that is not altered at any point in the code. It remains constant throughout, so the runtime should stay constant as well (O(10)).

- data: The mean runtimes are roughly constant, supporting my hypothesis that the Big O notation is O(N).

    iii. the length of the random text

hypothesis: I think that the run time will be O(N) since the start position is updated as we loop over the file.

2.

- MapGenerator

```
Varying k, using random text length 100 and file length 152145 (alice.txt)
k: 1    mean: 0.000266      stddev 0.000000     ci: [0.000266, 0.000267]
k: 2    mean: 0.000086      stddev 0.000000     ci: [0.000086, 0.000086]
k: 3    mean: 0.000144      stddev 0.000000     ci: [0.000144, 0.000144]
k: 4    mean: 0.000168      stddev 0.000000     ci: [0.000168, 0.000168]
k: 5    mean: 0.000144      stddev 0.000000     ci: [0.000144, 0.000144]
k: 6    mean: 0.000167      stddev 0.000000     ci: [0.000167, 0.000167]
k: 7    mean: 0.000196      stddev 0.000000     ci: [0.000196, 0.000196]
k: 8    mean: 0.000127      stddev 0.000000     ci: [0.000127, 0.000127]
k: 9    mean: 0.000200      stddev 0.000000     ci: [0.000200, 0.000200]
k: 10   mean: 0.000118      stddev 0.000000     ci: [0.000118, 0.000118]
k: 11   mean: 0.000125      stddev 0.000000     ci: [0.000125, 0.000125]
k: 12   mean: 0.000189      stddev 0.000000     ci: [0.000189, 0.000189]
k: 13   mean: 0.000206      stddev 0.000000     ci: [0.000206, 0.000206]
k: 14   mean: 0.000172      stddev 0.000000     ci: [0.000172, 0.000172]
k: 15   mean: 0.000101      stddev 0.000000     ci: [0.000101, 0.000101]

Varying text length, using k 5 and file length 152145 (alice.txt)
text length: 20      mean: 0.000026      stddev: 0.000000    ci: [0.000026, 0.000026]
text length: 40      mean: 0.000056      stddev: 0.000000    ci: [0.000056, 0.000056]
text length: 60      mean: 0.000075      stddev: 0.000000    ci: [0.000075, 0.000075]
text length: 80      mean: 0.000094      stddev: 0.000000    ci: [0.000094, 0.000094]
text length: 100     mean: 0.000128      stddev: 0.000000    ci: [0.000128, 0.000128]
text length: 120     mean: 0.000170      stddev: 0.000000    ci: [0.000170, 0.000170]
text length: 140     mean: 0.000123      stddev: 0.000000    ci: [0.000123, 0.000123]
text length: 160     mean: 0.000241      stddev: 0.000000    ci: [0.000241, 0.000241]
text length: 180     mean: 0.000238      stddev: 0.000000    ci: [0.000238, 0.000238]
text length: 200     mean: 0.000237      stddev: 0.000000    ci: [0.000237, 0.000237]
text length: 220     mean: 0.000274      stddev: 0.000000    ci: [0.000274, 0.000274]
text length: 240     mean: 0.000238      stddev: 0.000000    ci: [0.000238, 0.000238]
```

```
•  text length: 260    mean: 0.000291     stddev: 0.000000   ci: [0.000291, 0.000291]
•  text length: 280    mean: 0.000356     stddev: 0.000000   ci: [0.000356, 0.000356]
•  text length: 300    mean: 0.000421     stddev: 0.000000   ci: [0.000421, 0.000421]
•
•  Varying file length, using k 5 and text length 100
•  unique keys: 2694   mean: 0.000047     stddev 0.000000    ci: [0.000047, 0.000047]
•  unique keys: 2982   mean: 0.000044     stddev 0.000000    ci: [0.000044, 0.000044]
•  unique keys: 3939   mean: 0.000063     stddev 0.000000    ci: [0.000063, 0.000063]
•  unique keys: 7499   mean: 0.000058     stddev 0.000000    ci: [0.000058, 0.000058]
•  unique keys: 7777   mean: 0.000059     stddev 0.000000    ci: [0.000059, 0.000059]
•  unique keys: 28046  mean: 0.000104     stddev 0.000000    ci: [0.000104, 0.000104]
•  unique keys: 35722  mean: 0.000102     stddev 0.000000    ci: [0.000102, 0.000102]
•  unique keys: 41306  mean: 0.000123     stddev 0.000000    ci: [0.000123, 0.000123]
•  unique keys: 68922  mean: 0.000180     stddev 0.000000    ci: [0.000180, 0.000180]
•  unique keys: 143749        mean: 0.000157     stddev 0.000000    ci: [0.000157, 0.000157]
•
•  Finished tests
```

PART B:

3. /4.

i. Using default hash code will cause collisions because ".equals" is used in my boolean. Big O is O(1).

ii. Big O is O(N) because the mean runtimes are roughly constant.

iii. I hypothesized that the Big O time would be O(N^2), which was incorrect. Based on the data, the Big O time is log(n) for the TreeMap since the runtime is increasing logarithmically.

Data:

```
Starting tests

Varying k, using random text length 100 and file length 152145 (alice.txt)
k: 1   mean: 0.000223      stddev 0.000000     ci: [0.000223, 0.000223]
k: 2   mean: 0.000087      stddev 0.000000     ci: [0.000087, 0.000087]
k: 3   mean: 0.000102      stddev 0.000000     ci: [0.000102, 0.000102]
k: 4   mean: 0.000143      stddev 0.000000     ci: [0.000143, 0.000144]
k: 5   mean: 0.000125      stddev 0.000000     ci: [0.000125, 0.000125]
k: 6   mean: 0.000083      stddev 0.000000     ci: [0.000083, 0.000083]
k: 7   mean: 0.000119      stddev 0.000000     ci: [0.000119, 0.000119]
k: 8   mean: 0.000070      stddev 0.000000     ci: [0.000070, 0.000070]
k: 9   mean: 0.000068      stddev 0.000000     ci: [0.000068, 0.000068]
k: 10  mean: 0.000081      stddev 0.000000     ci: [0.000081, 0.000081]
k: 11  mean: 0.000081      stddev 0.000000     ci: [0.000081, 0.000081]
k: 12  mean: 0.000072      stddev 0.000000     ci: [0.000072, 0.000072]
k: 13  mean: 0.000071      stddev 0.000000     ci: [0.000071, 0.000071]
k: 14  mean: 0.000077      stddev 0.000000     ci: [0.000077, 0.000077]
k: 15  mean: 0.000083      stddev 0.000000     ci: [0.000083, 0.000083]

Varying text length, using k 5 and file length 152145 (alice.txt)
text length: 20     mean: 0.000017      stddev: 0.000000    ci: [0.000017, 0.000017]
text length: 40     mean: 0.000036      stddev: 0.000000    ci: [0.000036, 0.000036]
text length: 60     mean: 0.000065      stddev: 0.000000    ci: [0.000065, 0.000065]
text length: 80     mean: 0.000060      stddev: 0.000000    ci: [0.000060, 0.000060]
text length: 100    mean: 0.000067      stddev: 0.000000    ci: [0.000067, 0.000067]
text length: 120    mean: 0.000092      stddev: 0.000000    ci: [0.000092, 0.000092]
text length: 140    mean: 0.000126      stddev: 0.000000    ci: [0.000126, 0.000126]
text length: 160    mean: 0.000106      stddev: 0.000000    ci: [0.000106, 0.000106]
text length: 180    mean: 0.000128      stddev: 0.000000    ci: [0.000128, 0.000128]
text length: 200    mean: 0.000141      stddev: 0.000000    ci: [0.000141, 0.000141]
text length: 220    mean: 0.000145      stddev: 0.000000    ci: [0.000145, 0.000145]
text length: 240    mean: 0.000166      stddev: 0.000000    ci: [0.000166, 0.000166]
text length: 260    mean: 0.000249      stddev: 0.000000    ci: [0.000249, 0.000249]
text length: 280    mean: 0.000199      stddev: 0.000000    ci: [0.000199, 0.000199]
text length: 300    mean: 0.000213      stddev: 0.000000    ci: [0.000213, 0.000213]

Varying file length, using k 5 and text length 100
unique keys: 57     mean: 0.000042      stddev 0.000000     ci: [0.000042, 0.000042]
unique keys: 49     mean: 0.000043      stddev 0.000000     ci: [0.000043, 0.000043]
unique keys: 57     mean: 0.000055      stddev 0.000000     ci: [0.000055, 0.000055]
unique keys: 66     mean: 0.000060      stddev 0.000000     ci: [0.000060, 0.000060]
unique keys: 54     mean: 0.000058      stddev 0.000000     ci: [0.000058, 0.000058]
unique keys: 65     mean: 0.000071      stddev 0.000000     ci: [0.000071, 0.000071]
unique keys: 71     mean: 0.000074      stddev 0.000000     ci: [0.000074, 0.000074]
unique keys: 68     mean: 0.000069      stddev 0.000000     ci: [0.000069, 0.000069]
unique keys: 76     mean: 0.000068      stddev 0.000000     ci: [0.000068, 0.000068]
unique keys: 88     mean: 0.000074      stddev 0.000000     ci: [0.000074, 0.000074]

Finished tests
```