100 Boards (Times for 3 Trials and 4x4)

LexiconFirstAutoPlayer

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
6.474000	6.177000	12.720000	12.944000
6.707000	6.338000	12.802000	12.948000
6.431000	6.228000	12.146000	12.266000
6.537333	**6.247667**	**12.556**	**12.719333**

BoardFirstAutoPlayer

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
0.389000	0.286000	0.161000	0.173000
0.392000	0.213000	0.159000	0.163000
0.395000	0.235000	0.238000	0.230000
0.392	**0.244667**	**0.203**	**0.207**

1000 Boards (Times for 3 Trials and 4x4)

LexiconFirstAutoPlayer

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
61.789000	61.492000	119.550000	120.648000
66.578000	63.333000	123.330000	124.784000
64.752000	60.224000	121.122000	122.212000
64.373	**61.683**	**121.334**	**122.214667**

BoardFirstAutoPlayer

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
2.965000	1.942000	1.145000	1.217000
3.079000	2.135000	1.290000	1.30000
2.969000	2.039000	1.292000	1.303000
3.004333	**2.038667**	**1.242333**	**1.255333**

10000 Boards (4x4)

LexiconFirstAutoPlayer (took too long to do multiple trials)

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
634.359000	**616.727000**	**1220.232**	**1224.457**

BoardFirstAutoPlayer

SimpleLexicon	BinarySearchLexicon	TrieLexicon	CompressedTrieLexicon
18.878000	13.122000	7.544000	7.549000
19.492000	13.499000	7.893000	7.931000
19.611000	13.565000	7.959000	7.968000
19.327	**13.395333**	**7.798667**	**7.822333**

Throughout the data analysis, I ensured the reproducibility of the boards by using the following seed and keeping the minimum length of a word at 3.

BoggleBoardFactory.*setRandom*(**new** Random(12345));

CONCLUSIONS

1) When BoggleStats is run with the LexiconFirstAutoPlayer, the BinarySearchLexicon is the fastest. The data above supports this claim. For the tests done on 100, 1000, and 10000 boards, the fastest runtime always came from the BinarySearchLexicon. SimpleLexicon, TrieLexicon, and CompressedTrieLexicon came in second, third and fourth place, respectively, in terms of runtimes, across all three categories of tests.
2) The runtimes for all three lexicons were much faster because LexiconFirstAutoPlayer goes through many more iterations than BoardFirstAutoPlayer. Whereas BoardFirstAutoPlayer looks for possible words based on the letters in the boggleboard and stops upon dead ends, LexiconFirstAutoPlayer eliminates words one by one. As such, BoardFirstAutoPlayer is inherently faster than LexiconFirstAutoPlayer.

The BoardFirstAutoPlayer across all of 100, 1000, and 10000 board sizes show that TrieLexicon runs fastest out of the four lexicons.

Because TrieLexicon had the fastest runtime using BoardFirstAutoPlayer (and the fastest of all the combinations of Lexicons and AutoPlayer), I used it to find the best board (4x4 and 5x5) out of 50,000 searches.

Results:

Best 4x4 Board

I	I	N	O
L	M	T	O
I	A	E	S
D	N	R	T

Runtime –
34.764 Max –
1140
Total Board Count – 50000

Best 5x5 Board

P	A	C	O	D
O	X	S	E	R
A	T	N	T	R
N	I	E	A	S
D	R	N	C	E

Runtime – 109.370000
Max – 2318
Total Board Count – 50000

Estimating 100,000 and 1,000,000 game runtimes

LexiconFirstAutoPlayer:

The runtimes for each of the lexicons seem to increase by factors of ten for each of the four lexicons, based on the runtimes of 100, 1000, and 10,000 runs. For example, the runtime of SimpleLexicon is 6.537 for 100 boards, then 64.37 for 1000 boards, and finally 634.35 for 10,000 boards. The same change applies across the other three lexicons.

Predictions For LexiconFirstAutoPlayer:
100,000 games: ~6400 seconds
1,000,000 games: ~64000 seconds

BoardFirstAutoPlayer:

The runtimes for each of the lexicons seem to increase exponentially with the equation:

$$y=0.0035x^{0.841}$$

Predictions For BoardFirstAutoPlayer:
100,000 games: ~56 seconds
1,000,000 games: ~389 seconds