Callie Mao

**Compsci 201 - Boggle Analysis**

**1. Fastest Lexicon Implementation**

The below times were measured for LARGE\_LEXICON with iterative size 80612, word size 80612, and prefix size 16466:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SimpleLexicon** | **TrieLexicon** | **BinarySearchLexicon** |
| **Iteration Time (seconds)** | 0.014 | 0.224 | 0.001 |
| **Word Time (seconds)** | 0.009 | 0 | 0.001 |
| **Prefix (seconds)** | 0.033 | 0.034 | 0.023 |

The below times were measured for SMALL\_LEXICON with iterative size 19912, word size 19912, and prefix size 6143:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SimpleLexicon** | **TrieLexicon** | **BinarySearchLexicon** |
| **Iteration Time (seconds)** | 0.01 | 0.221 | 0.001 |
| **Word Time (seconds)** | 0.003 | 0 | 0 |
| **Prefix (seconds)** | 0.023 | 0.036 | 0.032 |

Based on the above data, BinarySearchLexicon is the fastest lexicon implementation. Its times are consistently lower than all times of SimpleLexicon and TrieLexicon measured with LARGE\_LEXICON and roughly less than all times of SimpleLexicon and TrieLexicon for SMALL\_LEXICON. The larger the number of words in the lexicon, the more efficient BinarySearchLexicon becomes relative to the other two methods, since it is able to halve its search options with each recursive call. BinarySearchLexicon becomes more efficient the larger the N, since BinarySearch usually has a runtime of O(logN) in relation to the size of the lexicon (a relationship that is not shared with SimpleLexicon and TrieLexicon) and thus becomes less efficient as N gets large.

**2**. **High-Score Board**

Based on 50,000 auto-game runs, the 4x4 and 5x5 boards giving the maximum score were generated to be the following:

c l i t p a c o d

s m e r o x s e r

b d a s a t n t r

c l e h n I e a s

d r n c e

**4x4 5x5**

Max Score: 1101 Max Score: 2120

**3. Time for Auto-runs**

For a 4x4 board, the following number of seconds were measured for the implementation of LexiconFirstAutoPlayer and BoardFirstAutoPlayer:

LexiconFirstAutoPlayer

|  |  |  |  |
| --- | --- | --- | --- |
| # Games | SimpleLexicon Time | TrieLexicon Time | BinarySearchLexicon Time |
| 10 | 0.542 | 1.152 | 0.467 |
| 500 | 24.465 | 48.053 | 24.327 |
| 1000 | 52.076 | 91.841 | 47.045 |
| 5000 | 245.715 | 449.786 | 271.637 |

BoardFirstAutoPlayer

|  |  |  |  |
| --- | --- | --- | --- |
| # Games | SimpleLexicon Time | TrieLexicon Time | BinarySearchLexicon Time |
| 10 | 0.126 | 0.101 | 0.101 |
| 500 | 0.886 | 0.956 | 0.817 |
| 1000 | 1.443 | 1.156 | 1.136 |
| 5000 | 7.963 | 3.246 | 4.481 |

For a 5x5 board, the following number of seconds were measured for the implementation of LexiconFirstAutoPlayer and BoardFirstAutoPlayer:

LexiconFirstAutoPlayer

|  |  |  |  |
| --- | --- | --- | --- |
| #Games | SimpleLexicon Time | TrieLexicon Time | BinarySearchLexicon Time |
| 10 | 0.816 | 1.489 | 0.705 |
| 500 | 36.007 | 63.798 | 33.453 |
| 1000 | 81.063 | 134.658 | 77.889 |
| 5000 | 392.339 | 707.702 | 412.933 |
| 10000 | 794.432 | 1301.489 | 808.166 |

BoardFirstAutoPlayer

|  |  |  |  |
| --- | --- | --- | --- |
| # Games | SimpleLexicon Time | TrieLexicon Time | BinarySearchLexicon Time |
| 10 | 0.126 | 0.101 | 0.101 |
| 500 | 0.886 | 0.956 | 0.817 |
| 1000 | 1.443 | 1.156 | 1.136 |
| 5000 | 7.963 | 3.246 | 4.481 |

Based on the above data, each of the different autoplayer and lexicon combinations result in an O(N) runtime. From this, it is not difficult to extrapolate how long 100,000 and 1,000,000 auto-runs will take. I use the equation of each line for each condition and plug in 100,000 and 1,000,000 auto-games to obtain the estimated time taken. The auto-runs will simply be a factor of the time that 10 auto-runs would take. 100,000 would be 10,000x the time of 10 auto-runs, while 1,000,000 runs would be 100,000x the time of 100 auto-runs.

Below are the predicted results for 100,000 games/trials for each of the auto-player and lexicon combinations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Time Prediction for 100,000 Auto-Games (in seconds)** | | |
|  |  | **SimpleLexicon** | **TrieLexicon** | **BinarySearchLexicon** |
| **4x4** | **LexiconFirstAutoPlayer** | 8,961.8284 | 4,930 | 5,400 |
| **BoardFirstAutoPlayer** | 160 | 70 | 90 |
| **5x5** | **LexiconFirstAutoPlayer** | 7,920 | 13,148 | 8,110 |
| **BoardFirstAutoPlayer** | 310 | 230.1419 | 180 |

Below are the predicted results for 1,000,000 games/trials for each of the auto-player and lexicon combinations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Time Prediction for 1,000,000 Auto-Games (in seconds)** | | |
|  |  | **SimpleLexicon** | **TrieLexicon** | **BinarySearchLexicon** |
| **4x4** | **LexiconFirstAutoPlayer** | 89,618.284 | 49,300 | 54,000 |
| **BoardFirstAutoPlayer** | 1600 | 700 | 900 |
| **5x5** | **LexiconFirstAutoPlayer** | 79,200 | 131,480 | 81,100 |
| **BoardFirstAutoPlayer** | 3,100 | 2301.419 | 1,800 |