Runtime Analysis

List Results

foxsays_tleader_in_50k.txt

```
Welcome to the profile statistics browser.
test_list.dat% Thu Sep  4 20:02:30 2025    test_list.dat

         176517 function calls in 39.758 seconds

   Random listing order was used

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1   39.654   39.654   39.754   39.754 cs412_foxsays_list.py:13(main)
    50000    0.034    0.000    0.067    0.000 cs412_foxsays_list.py:23(<lambda>)
        1    0.004    0.004   39.758   39.758 cs412_foxsays_list.py:1(<module>)
       37    0.000    0.000    0.000    0.000 <frozen codecs>:334(getstate)
      215    0.000    0.000    0.000    0.000 <frozen codecs>:322(decode)
        2    0.000    0.000    0.000    0.000 {method 'readline' of '_io.TextIOWrapper' objects}
        1    0.005    0.005    0.006    0.006 {method 'readlines' of '_io._IOBase' objects}
      215    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000   39.758   39.758 {built-in method builtins.exec}
        2    0.013    0.007    0.013    0.007 {built-in method builtins.print}
    50001    0.025    0.000    0.025    0.000 {method 'split' of 'str' objects}
        2    0.001    0.000    0.001    0.000 {method 'join' of 'str' objects}
    50002    0.011    0.000    0.011    0.000 {method 'strip' of 'str' objects}
    26036    0.011    0.000    0.011    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

foxsays_tleader_in_200k.txt

```
Welcome to the profile statistics browser.
test_list.dat% Thu Sep  4 21:44:56 2025    test_list.dat

         1584295 function calls in 5133.834 seconds

   Random listing order was used

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1 5132.563 5132.563 5133.809 5133.809 cs412_foxsays_list.py:13(main)
   200000    0.153    0.000    0.283    0.000 cs412_foxsays_list.py:23(<lambda>)
        1    0.025    0.025 5133.834 5133.834 cs412_foxsays_list.py:1(<module>)
     1405    0.000    0.000    0.000    0.000 <frozen codecs>:334(getstate)
     2114    0.001    0.000    0.009    0.000 <frozen codecs>:322(decode)
        2    0.011    0.006    0.019    0.010 {method 'readline' of '_io.TextIOWrapper' objects}
        1    0.019    0.019    0.020    0.020 {method 'readlines' of '_io._IOBase' objects}
     2114    0.008    0.000    0.008    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000 5133.834 5133.834 {built-in method builtins.exec}
        2    0.267    0.134    0.267    0.134 {built-in method builtins.print}
   200001    0.162    0.000    0.162    0.000 {method 'split' of 'str' objects}
        2    0.015    0.007    0.015    0.007 {method 'join' of 'str' objects}
   200002    0.042    0.000    0.042    0.000 {method 'strip' of 'str' objects}
   978648    0.568    0.000    0.568    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Dict Results

foxsays_tleader_in_50k.txt

```
Welcome to the profile statistics browser.
test_dict.dat% Thu Sep  4 20:01:33 2025    test_dict.dat

         176517 function calls in 0.123 seconds

   Random listing order was used

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.037    0.037    0.121    0.121 cs412_foxsays_dict.py:13(main)
    50000    0.027    0.000    0.057    0.000 cs412_foxsays_dict.py:20(<lambda>)
        1    0.002    0.002    0.123    0.123 cs412_foxsays_dict.py:1(<module>)
       37    0.000    0.000    0.000    0.000 <frozen codecs>:334(getstate)
      215    0.000    0.000    0.000    0.000 <frozen codecs>:322(decode)
        2    0.000    0.000    0.001    0.000 {method 'readline' of '_io.TextIOWrapper' objects}
        1    0.005    0.005    0.006    0.006 {method 'readlines' of '_io._IOBase' objects}
      215    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000    0.123    0.123 {built-in method builtins.exec}
        2    0.014    0.007    0.014    0.007 {built-in method builtins.print}
    50001    0.022    0.000    0.022    0.000 {method 'split' of 'str' objects}
        2    0.000    0.000    0.000    0.000 {method 'join' of 'str' objects}
    50002    0.011    0.000    0.011    0.000 {method 'strip' of 'str' objects}
    26036    0.005    0.000    0.005    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

foxsays_tleader_in_200k.txt

```
Welcome to the profile statistics browser.
test_dict.dat% Thu Sep  4 20:17:21 2025    test_dict.dat

         1584295 function calls in 101.077 seconds

   Random listing order was used

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1   99.839   99.839  101.039  101.039 cs412_foxsays_dict.py:13(main)
   200000    0.107    0.000    0.228    0.000 cs412_foxsays_dict.py:20(<lambda>)
        1    0.037    0.037  101.077  101.077 cs412_foxsays_dict.py:1(<module>)
     1405    0.000    0.000    0.000    0.000 <frozen codecs>:334(getstate)
     2114    0.001    0.000    0.009    0.000 <frozen codecs>:322(decode)
        2    0.011    0.005    0.019    0.009 {method 'readline' of '_io.TextIOWrapper' objects}
        1    0.019    0.019    0.020    0.020 {method 'readlines' of '_io._IOBase' objects}
     2114    0.008    0.000    0.008    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000  101.077  101.077 {built-in method builtins.exec}
        2    0.641    0.321    0.641    0.321 {built-in method builtins.print}
   200001    0.151    0.000    0.151    0.000 {method 'split' of 'str' objects}
        2    0.023    0.012    0.023    0.012 {method 'join' of 'str' objects}
   200002    0.042    0.000    0.042    0.000 {method 'strip' of 'str' objects}
   978648    0.196    0.000    0.196    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Dict Performance Improvement

The biggest waste of run time in the dict version of the program was creating the list of animals encountered based on the sounds heard. Originally for each sound encountered I was checking if the `animals_encountered` list already contained the animal. This is a poor implementation because it meant that for every sound encountered we were perform an O(n) operation (the `in` operator on list). To fix this I changed `animals_encountered` to be a set instead of a list. This alone was enough to speed up the run time of the 200K sample from ~101 seconds to ~1.8s (see below). The one drawback of this is that sets (in theory) do not preserve insertion order, and so we lose the order in which the animals were encountered.

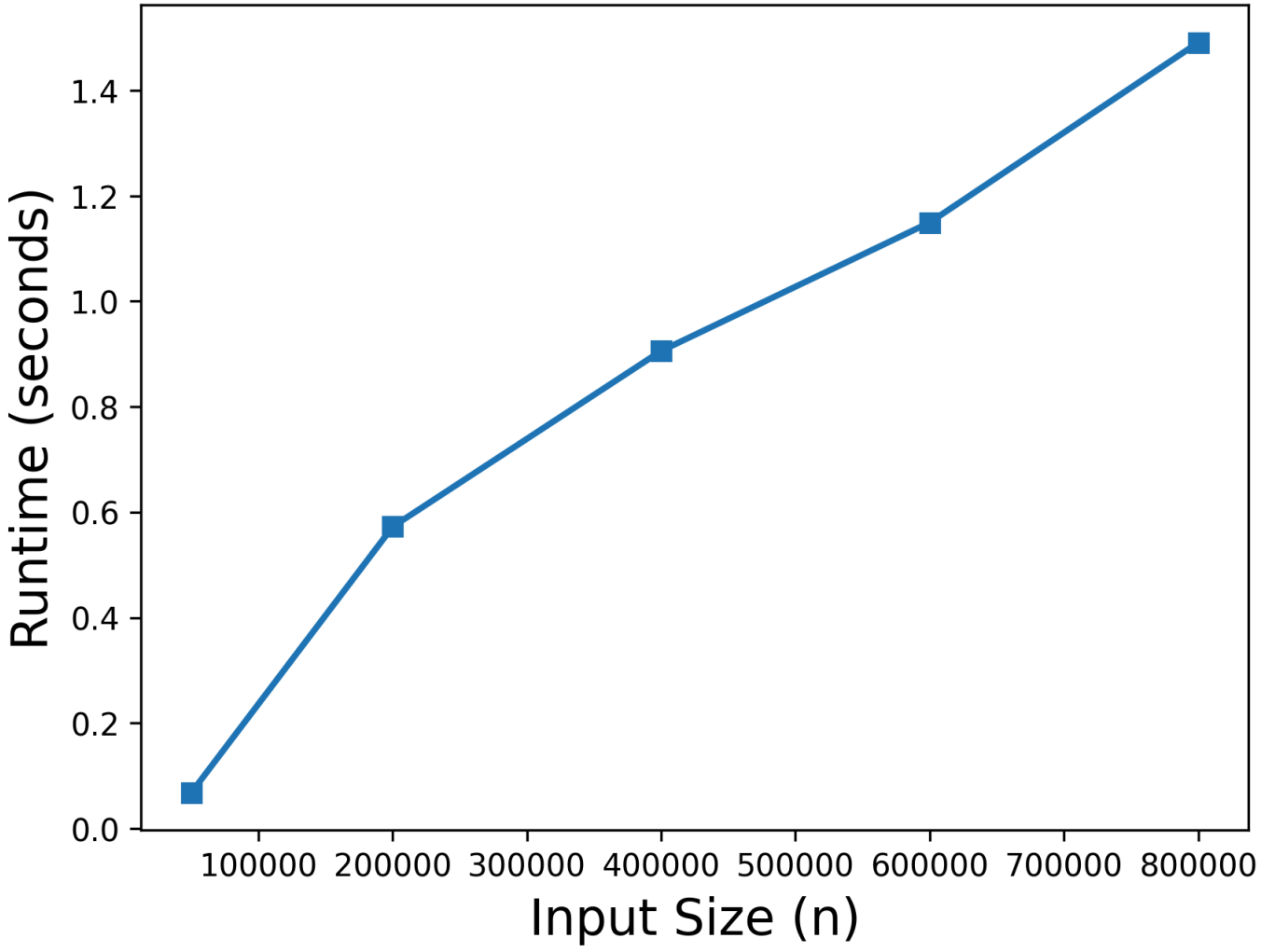
```
Welcome to the profile statistics browser.
test_dict.dat% Thu Sep  4 20:32:16 2025    test_dict.dat

         1384295 function calls in 1.742 seconds

   Random listing order was used

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.633    0.633    1.706    1.706 cs412_foxsays_dict.py:13(main)
        1    0.036    0.036    1.742    1.742 cs412_foxsays_dict.py:1(<module>)
     1405    0.000    0.000    0.000    0.000 <frozen codecs>:334(getstate)
     2114    0.001    0.000    0.009    0.000 <frozen codecs>:322(decode)
        2    0.011    0.005    0.019    0.009 {method 'readline' of '_io.TextIOWrapper' objects}
        1    0.019    0.019    0.020    0.020 {method 'readlines' of '_io._IOBase' objects}
     2114    0.008    0.000    0.008    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000    1.742    1.742 {built-in method builtins.exec}
        2    0.626    0.313    0.626    0.313 {built-in method builtins.print}
   200001    0.152    0.000    0.152    0.000 {method 'split' of 'str' objects}
        2    0.031    0.015    0.031    0.015 {method 'join' of 'str' objects}
   200002    0.045    0.000    0.045    0.000 {method 'strip' of 'str' objects}
    78487    0.018    0.000    0.018    0.000 {method 'add' of 'set' objects}
   900161    0.163    0.000    0.163    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Dict Runtime Plot



Based on the runtimes (computed with `time.perf_counter()`), there appears to be a relatively linear runtime for the dictionary based program relative to its input size. This is likely because we only have to iterate through every input line once.