CS51 Pset 5: Writeup

Kristine Zhang

3/29/2017

1 Code Overview

All code for testing can be found in the file performance.ml, and compiled/run by calling "ocamlbuild performance.byte" and "./performance.byte".

Crawling performance was tested by determining the amount of time it took to build an index for each of the three starter links, for each of the two set implementations.

Query performance was tested for all three of the starter links as well. This code determines how much time it takes to query each word in the dictionary, then returns the total time over all words.

The switch between the two set implementations was hardcoded by swapping DictSet and ListSet in myset.ml.

2 Crawling Performance

The results of the crawling tests are summarized in the table below:

Directory	DictSet Time (ms)	ListSet Time (ms)
simple-html	0.9449	1.045
html	366.5	383.2
wiki	62237	193701

Evidently, the DictSet implementation is faster than the ListSet implementation. The difference in speed becomes more obvious as the size of the directory increases; for example, crawling the wiki directory takes over 3 times as long using the ListSet implementation.

These results are unsurprising because the DictSet implementation is more efficient than the ListSet implementation for multiple purposes. Recall that the DictSet is implemented using a Binary Search Tree. Thus, operations like insert, remove, and member take $O(n)$ time when using ListSet, but only $O(log\,n)$ time when using DictSet.

3 Query Performance

The results of the query tests are summarized in the table below:

Directory	DictSet Time (ms)	ListSet Time (ms)
simple-html	0.003989	0.003890
html	0.1686	0.1173
wiki	23.48	19.70

There is no significant difference in the time taken for querying when using either the ListSet or DictSet implementation. This makes sense because querying does not actually depend on the implementation of the set itself. In either case, the query must process a user's request, search through a dictionary, and return the resulting set; however, the functions belonging to the set itself are never invoked.

It can be noted that the DictSet implementations seem to take slightly more time than the ListSet implementations. This could simply be due to variance, since the tests are run on relatively small samples, and the resulting timespans are small as well. Another possible explanation relates to the amount of space taken by a DictSet implementation. Recall that we implemented a DictSet by taking a Dict and setting all the values to Nil. Since the values are ignored, the DictSet takes more space than is strictly necessary. Returning this additional data could contribute to a small increase in the amount of time taken to query.

4 Conclusion

Overall, we see that the DictSet implementation is significantly faster than the ListSet implementation for crawling, but both approaches are similar for querying.