CS521 - Assignment 1

Michael Wathen

February 2, 2016

Collaborators

Ehsan Kermani helped me with the eunit test file as well as programming the `hw1:subtract` function. Mark Greenstreet helped me understand and implement the `hw1:search` function.

Question 1:

To run the tests

```
c(hw1).
c(hw1_test).
eunit:test(hw1_test).
```

Part a

See `nthtail` function given in file hw1.erl

Part b

`prefix` function given in file hw1.erl

Part c

See `prefix` function given in file hw1.erl

Question 2

Part a

See `libSubtraction` and `subPrint` functions given in file hw1.erl
See Table 1 for the results.

N	`lists:subtract`	time/(N^2)	`hw1:subtract`	time/$(N \log N)$	time/N
1000	17945	1.79e-02	128	1.85e-02	1.28e-01
2000	7446	1.86e-03	315	2.07e-02	1.58e-01
3000	17203	1.91e-03	537	2.24e-02	1.79e-01
5000	70788	2.83e-03	672	1.58e-02	1.34e-01
10000	203369	2.03e-03	1507	1.64e-02	1.51e-01
20000	721388	1.80e-03	2899	1.46e-02	1.45e-01
30000	1593127	1.77e-03	5146	1.66e-02	1.72e-01
50000	4423758	1.77e-03	6631	1.23e-02	1.33e-01

Table 1: Timing table for question 2. `lists:subtract` and `hw1:subtract` denote the elapsed time with the library function and my own (2c), respectively

Part b

From the results in Table 1 we see that the values in the time/(N^2) column in the table staysroughly constant. Therefore, the `lists:subtract` library function appears to be quadratic with respect to run time.

Without looking at the source code for `lists:subtract`, this behavior would be expected if every element of `L1` is checked by every element in `L2`. This would account for N^2 operations.

Part c

See `subtract` and `elementCheck` functions given in file hw1.erl

Part d

The results of my `subtract` function are given in Table 1. The timing results seem to be roughly $\mathcal{O}(N \log N)$ from columns time/$(N \log N)$ and time/N. We achieve this speed up in the `subtract` function since we order both lists to start with. This enables us to go through them together for comparison.

I'm working hard to try and understand `erlang` and functional programming so I apologise for extra length and poor `erlang` programming customs in this assignment.