CS521 - Assignment 1

Michael Wathen

February 8, 2016

Question 1:

See `move_par, q1_leaf1, q1_leaf2, q1_combine` functions

Question 2

See `rle, seq_rle` and `rle_helper`.

Question 3

See `longest_run` and associated functions

Question 4

a

See functions `best_match` and `firstOccur`, run `q4` to test function

b

P	N_1	N_2	T_s	T_p	T_s/(N_1*N_2)	T_p*P/(N_1*N_2)
4	201	15000	1.017e-1	3.995e-2	3.375e-8	5.300e-8
4	201	20000	1.531e-1	4.926e-2	3.809e-8	4.902e-8
4	201	25000	2.059e-1	9.395e-2	4.097e-8	7.479e-8
4	201	30000	2.200e-1	1.111e-1	3.648e-8	7.373e-8
4	201	35000	2.830e-1	1.296e-1	4.023e-8	7.366e-8
4	201	40000	2.752e-1	1.199e-1	3.423e-8	5.966e-8
4	201	45000	3.043e-1	1.179e-1	3.364e-8	5.216e-8
4	201	50000	3.316e-1	1.363e-1	3.299e-8	5.423e-8
4	201	100000	7.916e-1	2.451e-1	3.939e-8	4.877e-8

Figure 1: Timing tables for parts b and d, $T_{s,p}$ is the timing for the sequential and parallel algorithms, respectively.

See Figure 1 for timings. The order of the algorithm is $\mathcal{O}(N_1 N_2)$, this is reflected in the table.

c

See functions `best_match_leaf, best_match_combine, best_match_root` and `best_match_par`

d

See Figure 1 for timings. For large enough N_1, N_2 ($\approx (100, 1000)$) it seems that the parallel sequence match algorithm is better. However, when $N_1 > N_2$ then the sequential version of the algorithm is preferable.