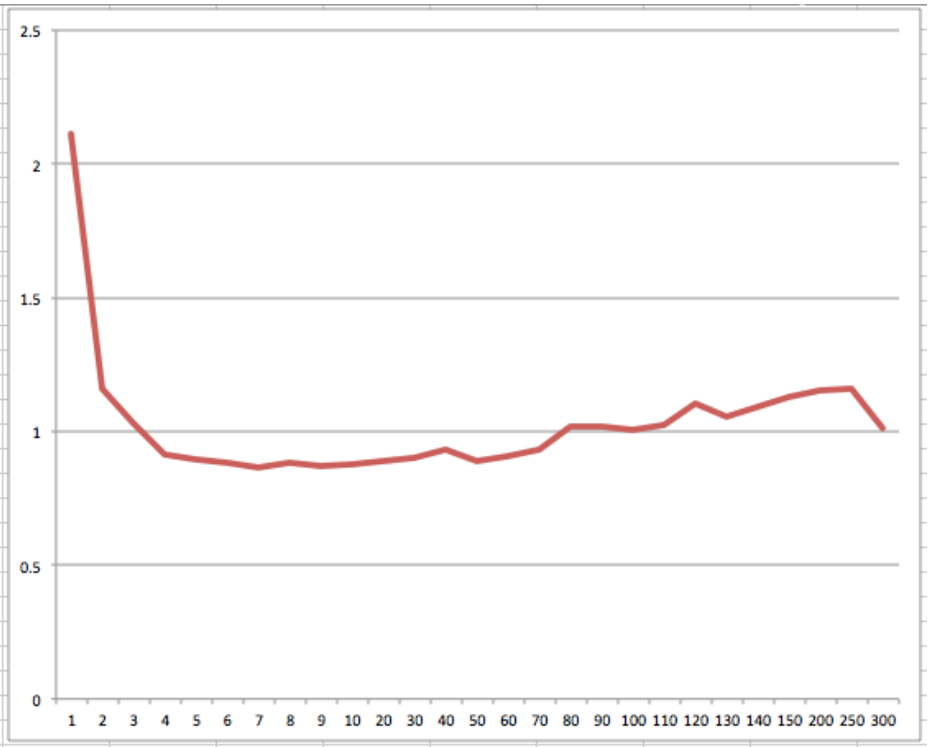
Report

JUNYI SUN
XUECHEN LI

1	2. 1144
2	1. 1575
3	1. 031
4	0. 9136
5	0. 893
6	0. 8852
7	0. 8652
8	0. 8861
9	0. 8719
10	0. 8758
20	0. 8904
30	0. 9033
40	0. 9321
50	0. 8895
60	0. 9103
70	0. 9345
80	1. 0175
90	1. 0169
100	1. 0091
110	1. 0276
120	1. 1048
130	1. 053
140	1. 0952
150	1. 1273
200	1. 1543
250	1. 1614
300	1. 0114



As we can see from the data, when n = 1, the running speed is very slow because we only have one process to work through all input files.

When we run the program in 2 processes, the program runs dramatically faster (approximately double the speed), because we split the work to more than one processes, they can work through different chunks of input files at the same time.

Then as we increase the number of processes up to about n = 7, the program decreases its running time.

When n = 7, the program has the best performance, which takes 0.8652 s to run the program.

When 5 < n < 20, the program runs in a relative fastest speed among all, with time at the range of 0.8s - 0.9 s.

Then when n > 150, as the number of processes increases, the run time is also increasing surprisingly, because running too many processes at the same time impacts the performance.