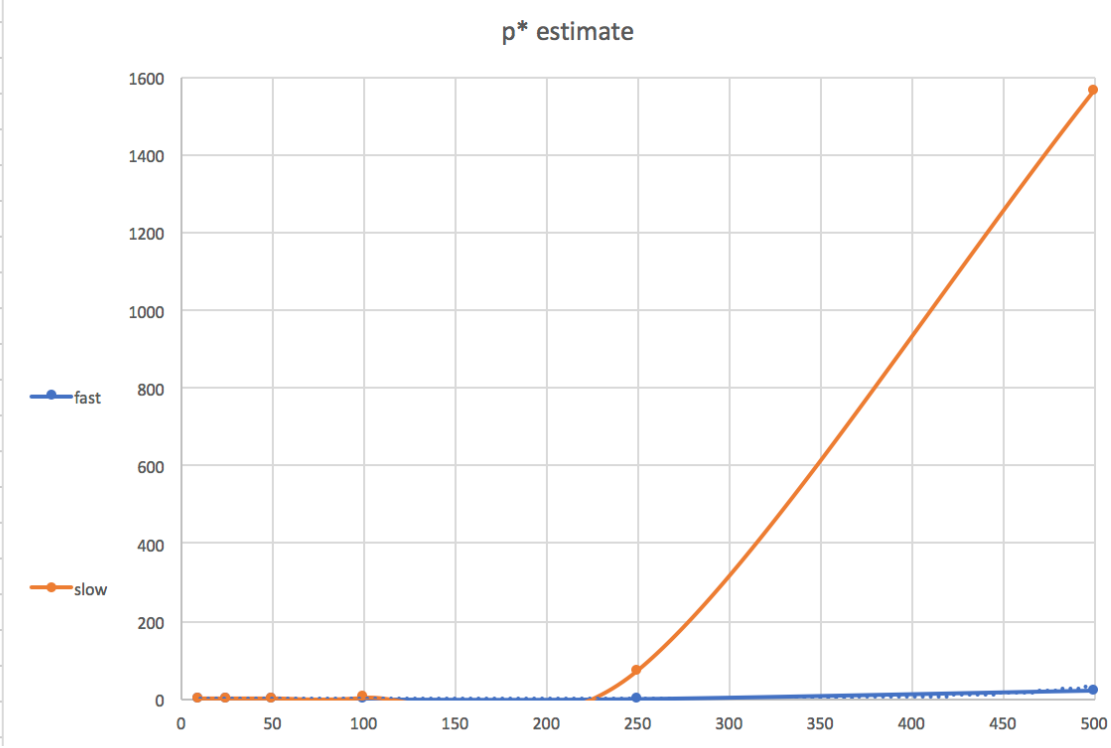
T = 30, N = 10,25 analyze : not much difference between slow and fast. Both used 0.006 second for total time, 0.570 threshold for fast, 0.597 for slow when N = 10, and 0.013 second and 0.586 threshold for fast, 0.02 second and 0.591 threshold for slow when N = 25.

T = 30, N = 50,100 analyze : Still not much difference, but slow UF starts get delayed than Weighted UF. Spent 0.044 second and 0.597 thresholds for fast, and 0.137 seconds 0.587 thresholds for slow when N = 50. Spent 0.13 seconds and 0.595 thresholds for fast and 1.701 seconds and 0.590 thresholds for slow when N = 100.

T = 30, N = 250, 500 analyze: shows Huge time gap between slow and fast. 1.214 seconds and 0.590 thresholds for fast and 72.785 seconds and 0.593 thresholds for slow when N = 250. And when N = 500, it shows more extremely, 20.243 seconds and 0.594 thresholds for fast, and 1563.552 seconds and 0.591 thresholds for slow.

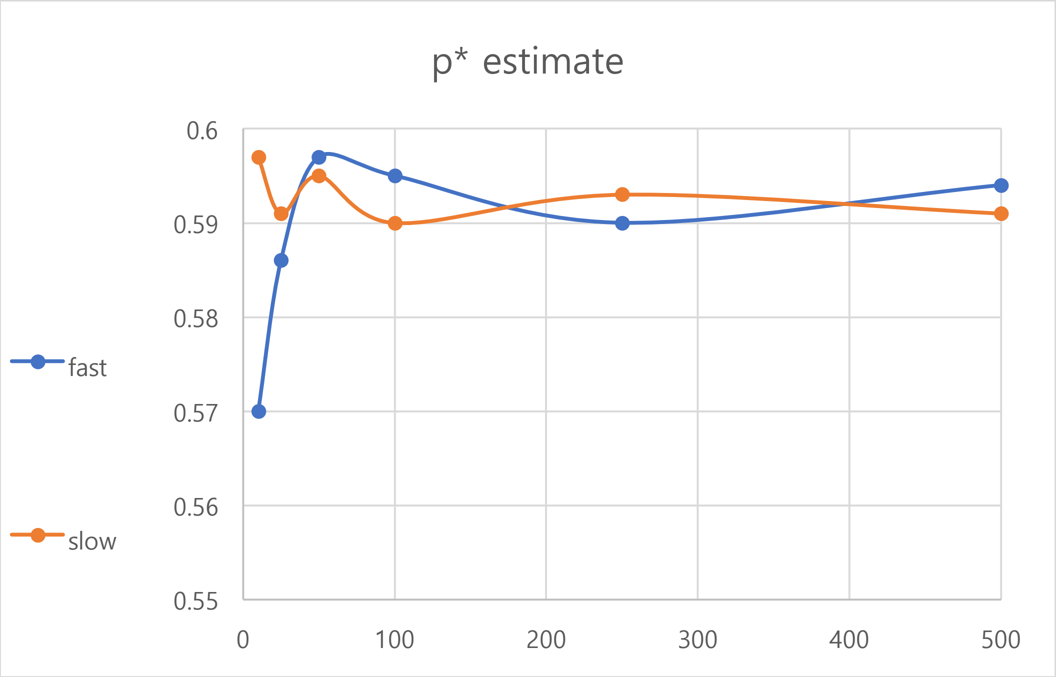
Running time table (in seconds)

n	Fast	Slow
10	0.006	0.006
25	0.012	0.02
50	0.044	0.137
100	0.13	1.701
250	1.214	72.785
500	20.243	1563.552



P* estimate table

N	fast	slow
10	0.57	0.597
25	0.586	0.591
50	0.597	0.595
100	0.595	0.59
250	0.59	0.593
500	0.594	0.591



Overall : Time difference between slow and fast UF goes extremely when N goes more higher, but Threshold keeps nearby 0.570-0.590, it doesn't exceed too much or too less. standard deviation of time for fast is stable, but standard deviation of time for slow is getting bigger because time goes too high because of running time. After testing N over 250, I could see and feel that Weighted Quick Union is extremely faster than Quick Union. It shows the basic thinking that why time complexity is so important when code the program. It has same function but Weighted Quick Union can do it fast when Quick Union spends forever.