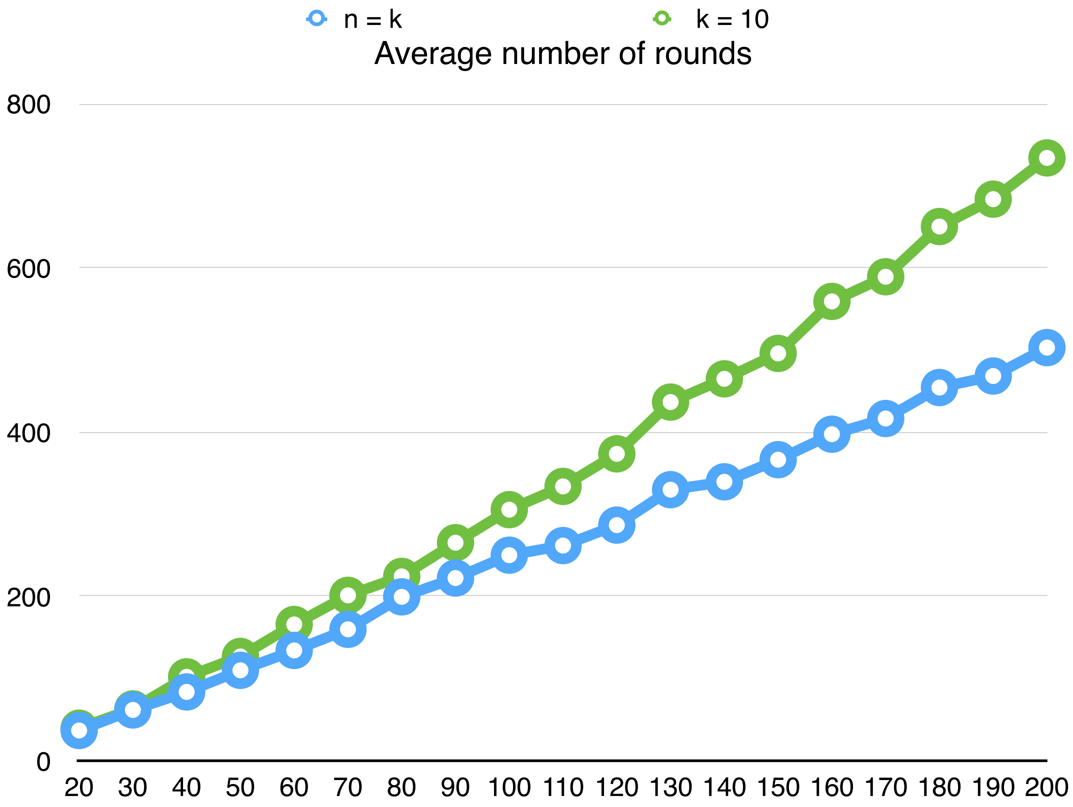
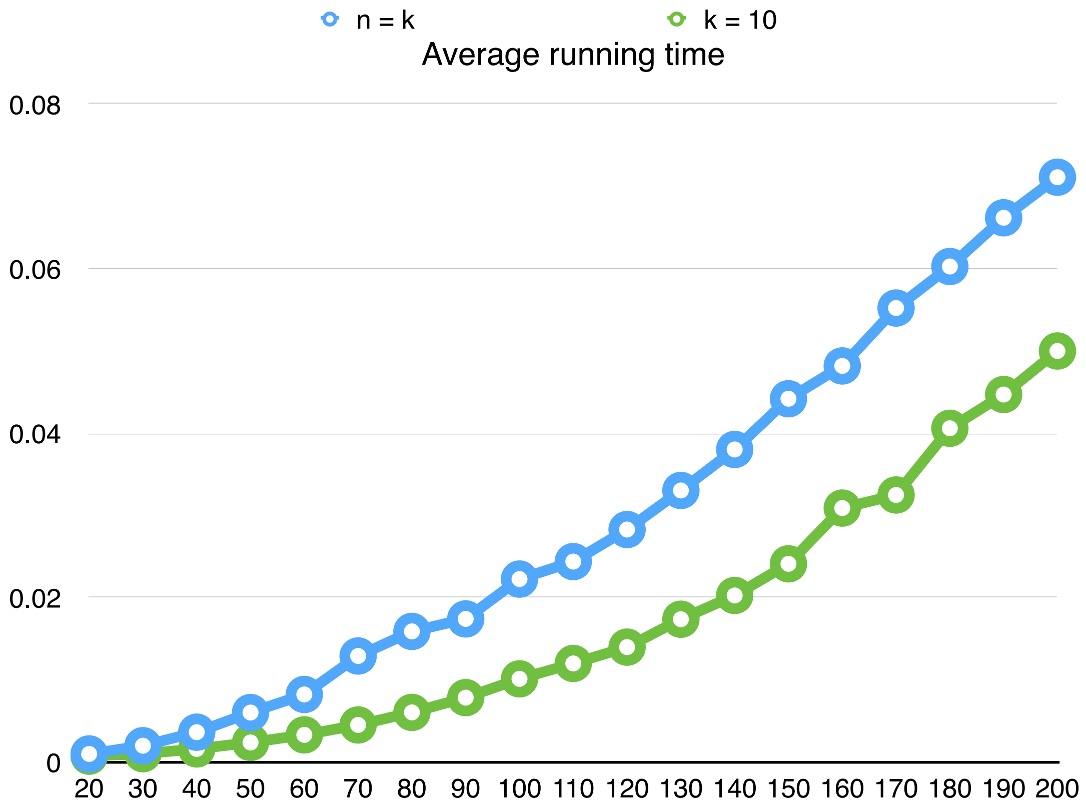
n = k	Average number of rounds	Average time (s)
20	36.9	0.0011571094999965226
30	61.9	0.0023574123000003055
40	83.8	0.0036456443999952626
50	110.1	0.0060318759999944405
60	134.3	0.008231582000001935
70	160.0	0.012913058100005514
80	199.5	0.01589963429999841
90	222.4	0.017420513100003632
100	250.2	0.022264650800002526
110	261.9	0.024390191899999535
120	286.8	0.028289753799995765
130	329.9	0.03304024399999435
140	339.5	0.038024863300009315
150	366.4	0.04419520560000763
160	397.4	0.04818456770000523
170	416.5	0.05522614250000117
180	454.2	0.06027623529999175
190	468.2	0.0662024187999947
200	502.8	0.07113094379998301

n, k =10	Average number of rounds	Average time (s)
20	39.7	0.0007308204999844747
30	63.2	0.0010315548999756175
40	102.2	0.001612083200006964
50	127.0	0.0023942870000041696
60	165.9	0.003319522300000699
70	201.3	0.004525537299980442
80	224.4	0.00605090259998633
90	265.4	0.007846986800007016
100	305.6	0.010132355699988693

110	333.9	0.012002835799989953
120	373.5	0.013988091699968664
130	436.5	0.017380392199981998
140	464.8	0.020271018199991886
150	495.7	0.024119004999977278
160	559.0	0.03089774310000166
170	589.0	0.03249262520000684
180	650.1	0.04060023639999599
190	683.4	0.04470685650001087
200	733.6	0.05038815170000817



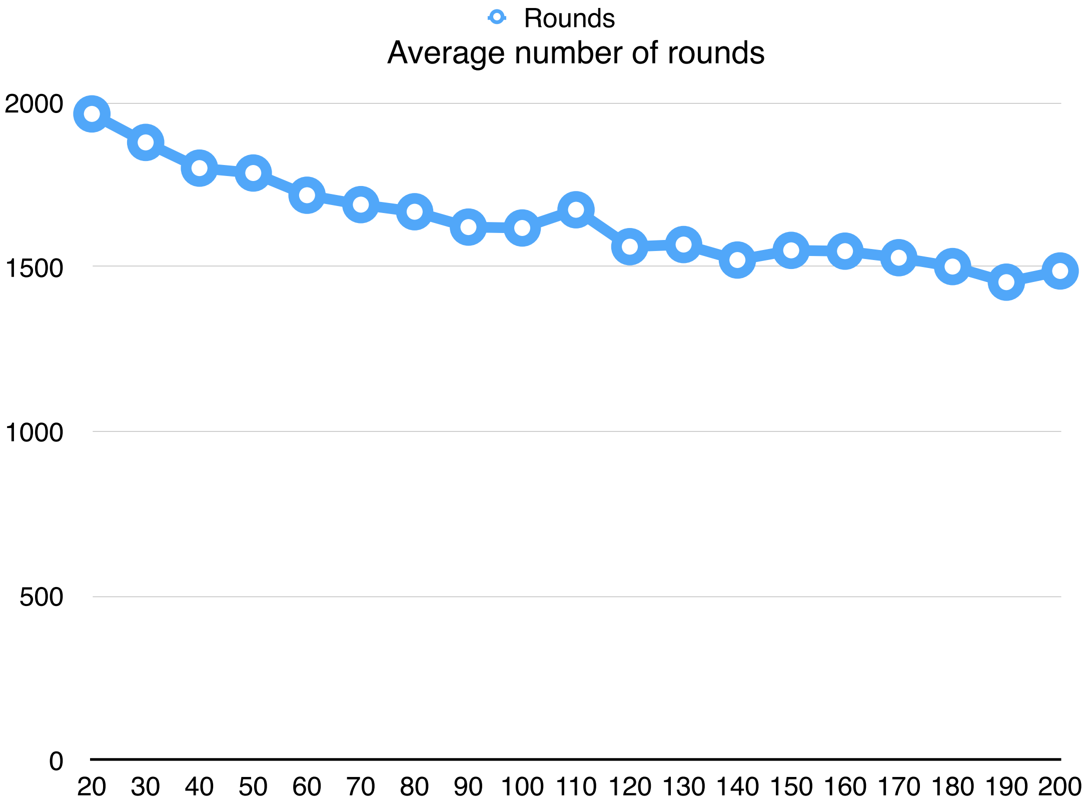
The table and graphs from above shows that when n = k, the program requires less number of rounds to complete the task, however, has a longer running time. When k is constant at k = 10, the program requires more number of rounds to complete the task at a shorter running time.

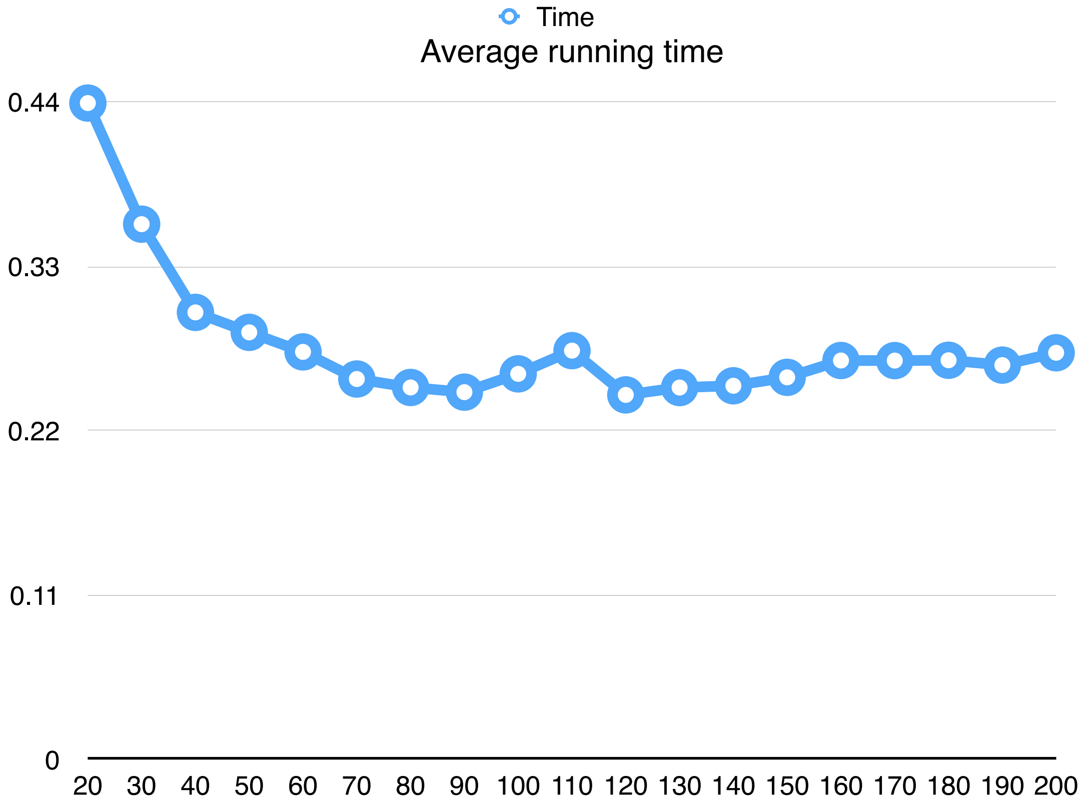


n = 500, k	Average number of rounds	Average time (s)
20	1965.7	0.43867060890001996
30	1879.0	0.3577333881999493
40	1800.8	0.2994789141000183
50	1785.9	0.28564462449999156
60	1718.3	0.27257980849999514
70	1689.7	0.25444607739991626
80	1668.1	0.24878616069991039
90	1621.4	0.24564862520001043
100	1618.7	0.25787634099997375
110	1674.1	0.27341947099994285
120	1561.8	0.24382135460000426
130	1568.3	0.24873213760006366
140	1521.2	0.24991283899994415
150	1550.6	0.255562000999862
160	1548.2	0.2668189597000037
170	1528.3	0.26676241939999273
180	1501.3	0.2669802526000694

190	1453.9	0.2637430179999683
200	1487.9	0.27178978030001416

From the data above, when n is constant at n = 500. As k increases the number of rounds decreases as well. There is a significant drop in the average running time between n = 20 to n = 60, and then stays between the range of approximately 0.25 to 0.27 for the rest of the tests.





By restricting the number of preferences allowed in the algorithm, many hospitals/students individuals are forced to share the same preferences lists, this means that a lot of reassigning occurs and this increases the number of rounds. With a larger number of k, there is a higher chance of having distinctive preferences hence, smaller chance of having a clash of preferences. A larger number of k also means that the program needs to loop through more unique lists when finding on a stable partner. Hence increases the average running time.

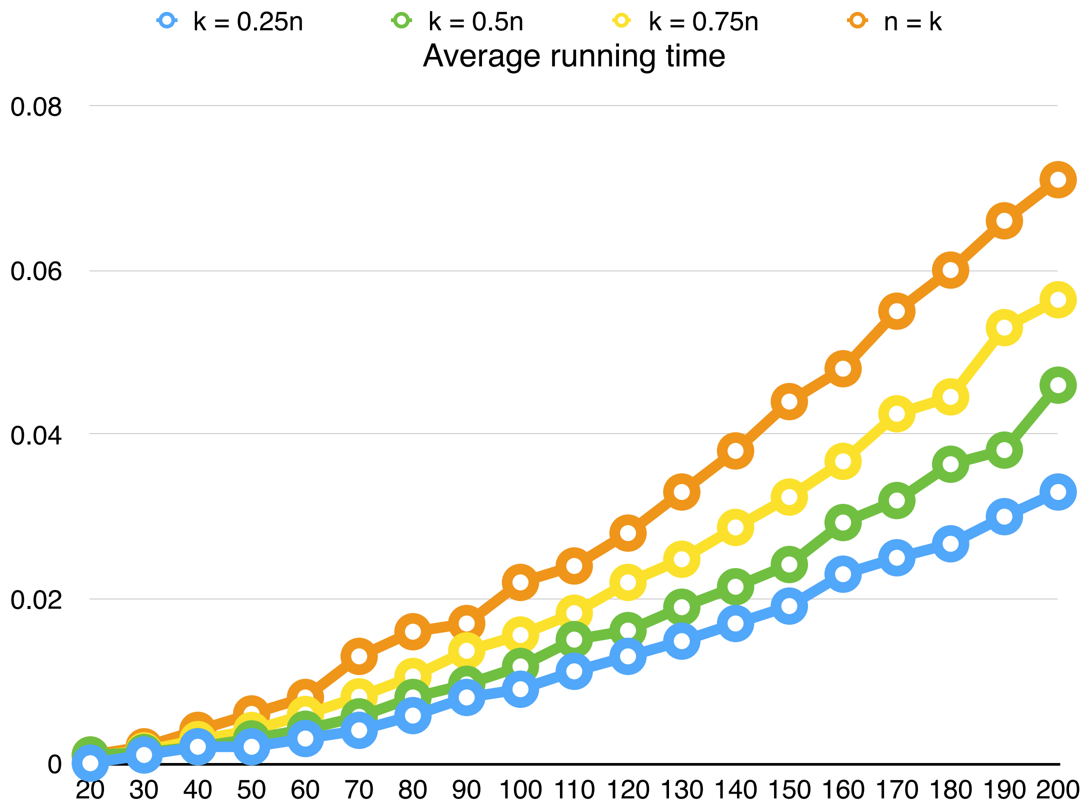
Restriction: Use k = n as control, let k= 0.25n, k = 0.5n, k = 0.75n for n = [20, 200] in intervals of 10.

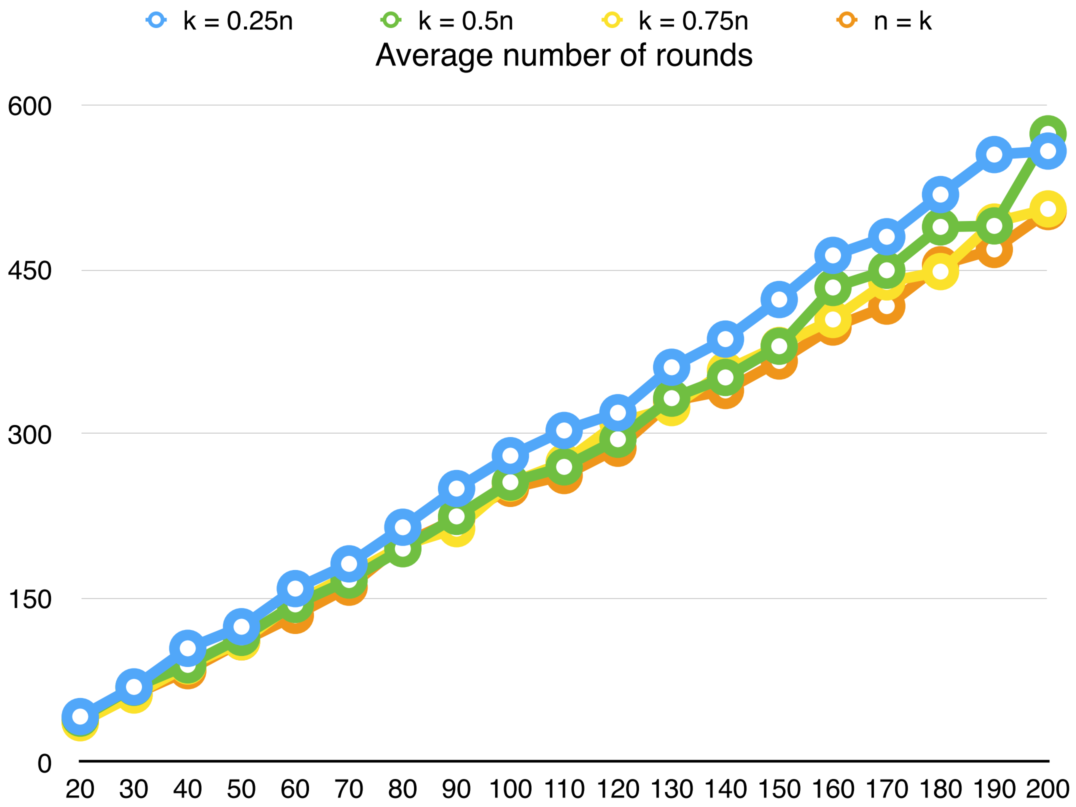
n	k = 0.25	Average number of rounds	Average time (s)
20	5	41.9	0.0004912389998935396
30	7	69.0	0.0010261298999466816
40	10	104.2	0.00161527490017761
50	12	123.9	0.0022459529003754143
60	15	158.7	0.0031718668999019426
70	17	181.3	0.004153344799669867
80	20	214.7	0.005794308700023976
90	22	250.0	0.008281934699880367
100	25	280.0	0.008743177600081253
110	27	303.0	0.011181639400274435
120	30	319.2	0.013279301399961696
130	32	360.7	0.014852486400013732
140	35	386.4	0.017038711199711544
150	37	422.5	0.019117628099957075
160	40	462.8	0.02330677089994424
170	42	479.8	0.024519588900147937
180	45	518.3	0.02669927679999091
190	47	554.8	0.029898424700331817
200	50	558.0	0.0333585126003527

n	k = 0.5	Average number of rounds	Average time (s)
20	10	40.5	0.0008379900999898382
30	15	68.6	0.0013033877998168465
40	20	89.1	0.002011094600220531
50	25	114.2	0.002958472399950551
60	30	144.3	0.0041318845000205325
70	35	166.5	0.005644037099864363
80	40	195.1	0.00756153259999337
90	45	224.6	0.009600594499897853
100	50	255.8	0.011772598400057177
110	55	269.8	0.015037747400037915
120	60	295.0	0.01613795930006745
130	65	332.5	0.018980377999832855
140	70	351.4	0.021471337499770017
150	75	379.9	0.024178879899773166
160	80	433.5	0.029284559999814518
170	85	449.2	0.03196259999995164
180	90	488.8	0.03637851139992563
190	95	489.7	0.03809396170008768
200	100	573.7	0.0457408044998374

n	k = 0.75	Average number of rounds	Average time (s)
20	15	36.3	0.0008698691000972758
30	22	61.6	0.001511139000103867
40	30	88.1	0.0028728345998388248
50	37	110.2	0.00397959499987337
60	45	146.2	0.005927419399995415
70	52	171.1	0.008082029499928468
80	60	198.2	0.010557425500155659
90	67	213.4	0.013676444100201478
100	75	255.1	0.015600576900214946
110	82	273.9	0.018255633800254144

120	90	310.2	0.022038969499590166
130	97	323.9	0.024818581200270272
140	105	357.8	0.028689495899561733
150	112	380.6	0.0323844549997375
160	120	404.3	0.0367312830998344
170	127	438.6	0.042517370499990645
180	135	448	0.044574906600064426
190	142	493.2	0.05260705130021961
200	150	505.2	0.05639264539950091





In conclusion, the data above shows the average number of rounds for each input are very close with k = 0.25n growing the fastest. However, there is a relatively large difference in average running time as k inputs increase. As k increases, the chances of similar preferences between n hospital/students decreases and less swapping occurs. This decreases the number of rounds but increases the average running time as the program needs to loop through more unique lists.