Lab 1 Report

System Values

- Command used was lscpu

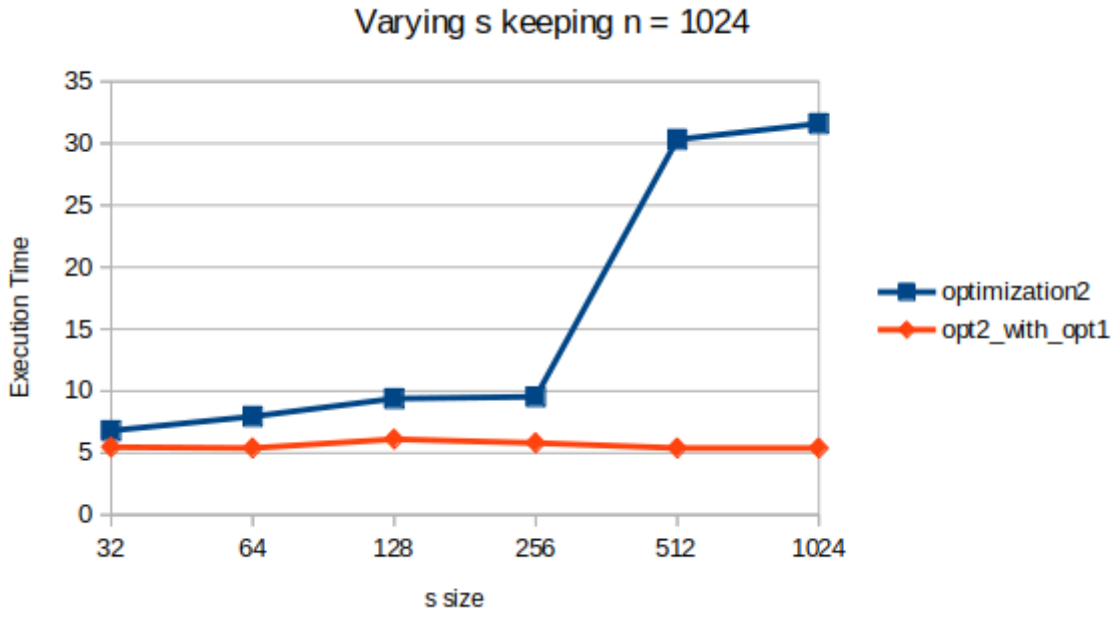
- Cache size
  L1 cache - 32 KB
  L2 cache - 256 KB
  L3 cache - 6144 KB
- Avg Clock Speed – 2100 MHZ
- Clock time – 0.476 ns

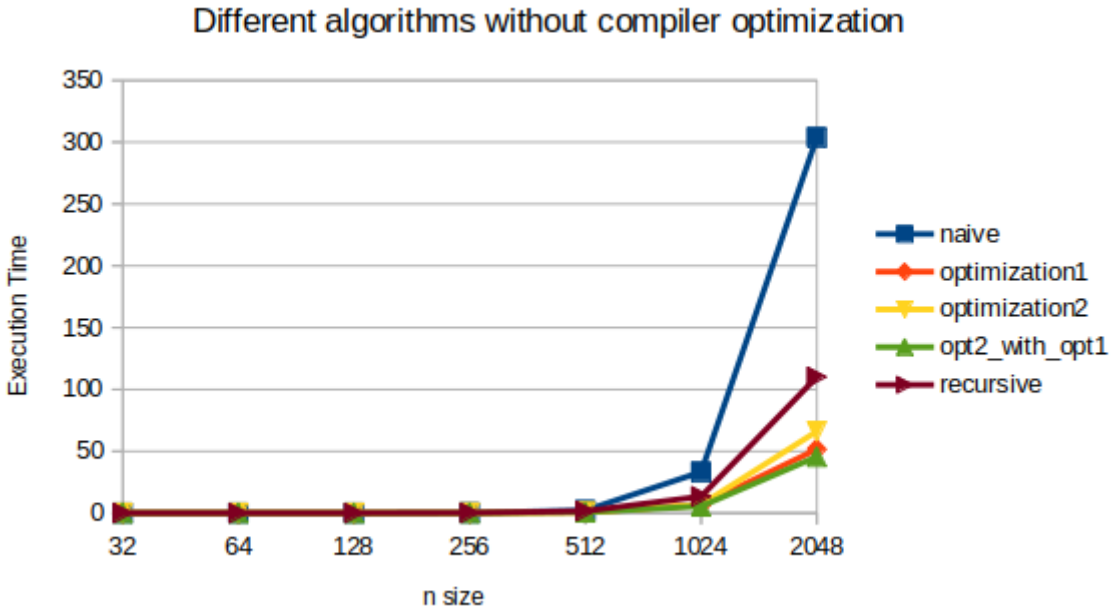
Cache Size L1 =32 KB and int size= 4B.
So, matrix size that can fit in cache = sqrt(32*2^10/4)=90.5.

For most modern processors the standard integer operations take just one clock cycle, excepting just multiplication and division (if available). Multiplication typically takes around 6 cycles

For every element of the resultant matrix n-1 addition operation is required and n multiplication
Therefore,
theoritical value of the time taken by code =((n-1) * (time taken in one addition operation between two integer) + n **(time taken in one multiplication operation between two integer))***n*n
=n*n*(n-1)*0.476 + n*n*n*6*0.476
=3.332n^3 -0.476*n^2
All time in miliseconds

n	Theoritical time
64	0.871514
128	6.979911
256	55.870488
512	447.089
1024	3577.209
2048	28619.665



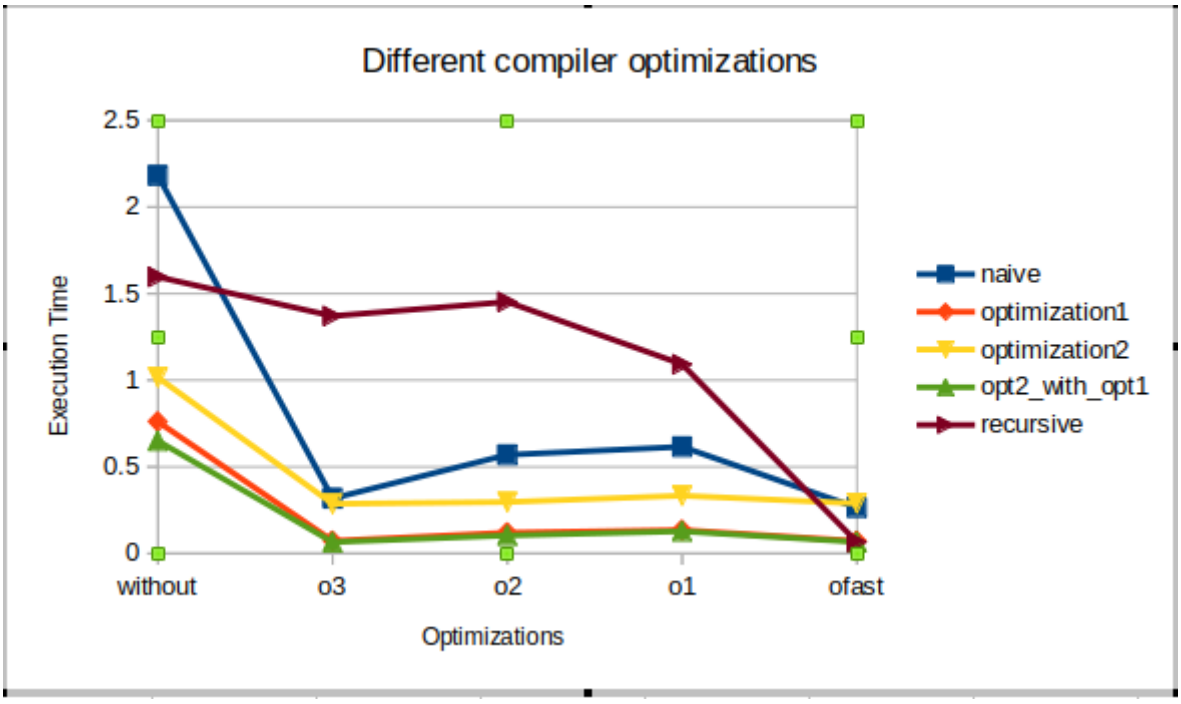


S used over here is 32

- Naive algorithm is the slowest as it faces many cache misses in comparison to other methods
- Optimization 1 works better than naive since it accesses all the elements in a row of second matrix at a time to compute an element of the resultant matrix.
- Optimization 2 works better than naive since it involves matrix multiplications in blocks

n	naive	optimization1	optimization2	Optimization2with1	recursion
32	0.001043	0.000969	4.46E-04	5.12E-04	0.000393
64	0.008518	0.010013	6.12E-03	4.51E-03	0.020455
128	0.040673	0.012519	1.05E-02	1.06E-02	0.050318
256	0.204321	0.093092	0.080468	0.084414	0.231504
512	2.20181	0.680257	0.734921	0.703443	1.6363
1024	33.2576	6.56104	6.80322	5.47096	13.412188
2048	303.84	51.5604	66.1257	45.6008	110.124673

All time unit is in sec



Over here n=512, s=256

- Results observed are as follows:

	naive	optimization1	optimization2	Optimization2with1	recursion
Normal	2.20181	0.680257	1.01613	0.803234	1.6363
O3	0.420671	0.093035	0.316316	0.067541	1.300214
o2	0.604463	0.292576	0.302918	0.113564	1.462512
O1	0.720561	0.253489	0.332552	0.128901	1.102137
Os	0.30153	0.081363	0.301803	0.069426	1.329535
Ofast	0.235383	0.069205	0.291023	0.071629	0.06536