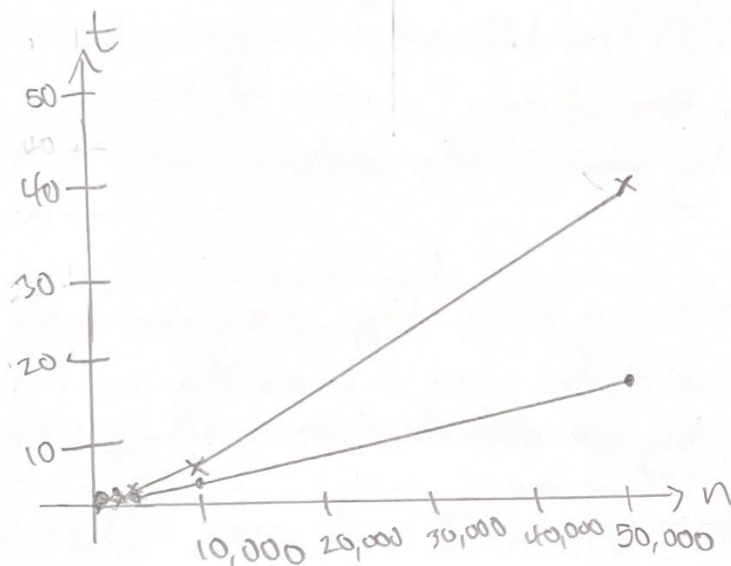


## Part 1

#rows	avg time row-wise	avg time col-wise
10	.000017	.000018
100	.000097	.000086
1000	.006966	.007150
10,000	.704850	1.084665
50,000	17.651499	39.244868



$n = \# \text{ rows}$

$t = \text{avg excc time}$

●● = row-wise

xx = col-wise

1. The concept of spatial locality refers to the relative proximity of data accessed by a program. In general programs should access addresses near other recently accessed data. Good spatial locality clusters memory by usage. Bad spatial locality has require the data at disparate addresses.
2. The row-wise should be more performant than col-wise. In row-wise access the memory will be iterated through sequentially along the contiguous blocks of memory allocated to each row. However in col-wise access the cursor will jump across arrays for every outer loop, giving poorer spatial locality.

3. The disparity between the performances grows as the array size increases. The increase in size increases the gaps across which col-wise must leap every time it jumps between row arrays. Thus row-wise is increasingly faster as it has better spatial locality.
4. Yes. Can't really stretch this one out.
5. As previously stated in (3) the difference in execution times is primarily due to the increase in size of the row arrays. This significantly worsens the spatial locality of the col-wise access.
6. The execution time varies every time the program runs. Had we only considered a single run the results in (3) could have been skewed by an outlier. We average the results over 10 runs to minimize the chances of this happening.



### Part 3

SZ	col-wise exec.		row-wise exec	
	hits	misses	hits	misses
10		4,620		4,558
100		4,967		4,615
1,000		9,126		6,224
10,000		118,094,814		132,557
50,000		5,249,620,366		3,092,090

1. I believe the command I was meant to use was

\$ perf stat <sup>or, /cw</sup> ./rw

However this command did not correctly display the cache hits or misses. Rather, I was told "some events weren't counted, try disabling the NMI watchdog". However, since I lacked the permissions to do this (I tried), I was forced to use

\$ perf stat -e cache-misses <sup>or, /cw</sup> ./rw

I could not find an analogous command for cache hits.

2. A cache hit occurs when a cache has the requested data locally within its memory. A miss occurs when this data cannot be found locally (i.e. at the  $k^{th}$  level) and must be searched for (e.g.  $k+1$ ). Here the cache misses are much higher for col-wise, indicating worse performance

3. The spatial locality of a program can be inferred from the ratio of hits to misses. A program with many hits and few misses can be said to have high spatial locality. This can be easily observed by the repeated trends in both tables showing col-wise to have worse spatial locality