BGN: 2191A

P5

Q7

a.

i. A[i], B[i], and B[i][0] experience temporal locality. Temporal locality is when the same resource is accessed repeatedly within a span of time. Since "i" only changes every 1024 iterations of the inner loop, A[i], B[i], and B[i][0] refer to the same resource for 1024 consecutive accesses.

ii. A[i][j] and A[j] experience spatial locality. Spatial locality is when resources close together in memory are accessed within a short amount of time of one another. In this case, since "j" is incremented by 1 every inner-loop iteration, and arrays in C are stored contiguously in memory, A[i][j] are stored at adjacent locations in memory from one such iteration to the next, as are A[j].

b.

Tag	**Index**	**Offset**	**Hit/Miss Status**
0x0	0x0	0x3	Miss
0x0	0xb	0x4	Miss
0x0	0x0	0x1	Hit
0x1	0x0	0x1	Miss
0x0	0x0	0x1	Miss
0x2	0xc	0x2	Miss
0x0	0x0	0x4	Hit
0x2	0xc	0x0	Hit

c. A data hazard is when an instruction requires some data (e.g., values in some register) which are not available yet, as they are still being calculated/have not been stored yet by a previous instruction.

Load instructions can be reordered to eliminate read-after-write dependencies. If a value from memory is written by an instruction, that should not be immediately followed by an instruction which reads that same address.

Arithmetic instructions can be reordered to remove consecutive pairs of instructions where the latter depends on a calculation performed by the prior (e.g., the result of an addition is stored in a register, and the next instruction uses that register's value)

d. The dirty cache line which takes up the space in which the data being written should go is written back to the L2 cache (or main memory, depending on whether there is an L2 cache) and the data to be written is put in that line's place in the L1 cache, which is still dirty.