CS230
Quiz-8 Solutions

Soln 1:
Ans: C

Explanation:

Number of sets in cache = v. So, main memory block j will be mapped to set (j mod v), which will be any one of the cache lines from (j mod v) * k to (j mod v) * k + (k-1). (Associativity plays no role in mapping- k-way associativity means there are k spaces for a block and hence reduces the chances of replacement.)

Soln 2: 0.9, 0.88, 0.89

Ans: 0.9 (if both levels of memory are accessed simultaneously)

Explanation:

Cache access time-> T_1
Level-2 memory access time-> T_2

Given Data:

T_{avg} = 20 ns

T_1 = 10 ns

T_2 = 100 ns

Formula:

$T_{avg} = H*T_1 + (1 - H)(T_1 + T_2)$

$= T_1 + (1 - H)(T_2)$

Calculation:

$20 = 10 + (1 - H)100$

$\therefore H = 0.9$

Hit ratio = 0.9

Explanation:(if accessing memory level wise i.e hierarchical)

Formula:

$T_{avg} = H*T_1 + (1 - H)(T_2)$

Calculation:

$20 = 10H + (1 - H)100$

$\therefore H = 8/9$

Hit ratio = 0.88 OR 0.89

Soln 3:

Ans: 0.5 (if considering read and write request of A[i][j] as a single access) ,

0.75(considering both read and write as different accesses)

0.75 is the correct answer

Explanation:**(for 0.5)**

Given that column major order is used:

Thus consecutive memory locations (words) store elements like this: A[0][0], a[1][0], A[2][0].....etc.

One element is of 4 Bytes and one cache block is of 8 Bytes size.

So whenever a cache miss occurs, consecutive 8 Bytes ,(ie 2 elements) are fetched and placed in cache.

In given code

```
for(i=0; i<10; i++)
   for(j=0; j<10; j++);
```

A[i][j] = A[i][j]+10;

When i=0,j=0

In A[i][j] = A[i][j]+10;

A[0][0] is miss -> A[0][0] is searched in memory and fetched consecutive 8 Bytes,

ie, A[0][0] and A[1][0] are in memory.

Now while writing A[i][j] = A[i][j]+10;

cache hit occurs because A[0][0] is already in cache

But next i=0 and j=1 and we need A[0][1].

But we have A[0][0] and A[1][0] in cache.

Again cache miss occurs.

Thus while reading every new element cache miss occurs. While writing back cache hits.

Each loop contains one read (always miss) and one write (always hit).

Total 10*10 =100 times loop works.

Thus 2*100=200 memory references and 100 reads and 100 writes

cache hit ratio = no of hits / total no of accesses. = no of writes/ no of accesses = 100/2*100 = 1/2

0.75 (considering both read and write)

Explanation:

First time when the statement “A[i][j] = A[i][j] + 10” is executed. (A[0][0] = A[0][0] + 10) We get read miss but write hit. As column major order A[0][0] will bring A[1][0] with it in the cache. So for A[1][0] both write and read hit. So for two elements A[0][0] and A[1][0] the hit ratio will be (2 write hits + 1 read hit) / (4 accesses) = 0.75. The same pattern follows for remaining elements.