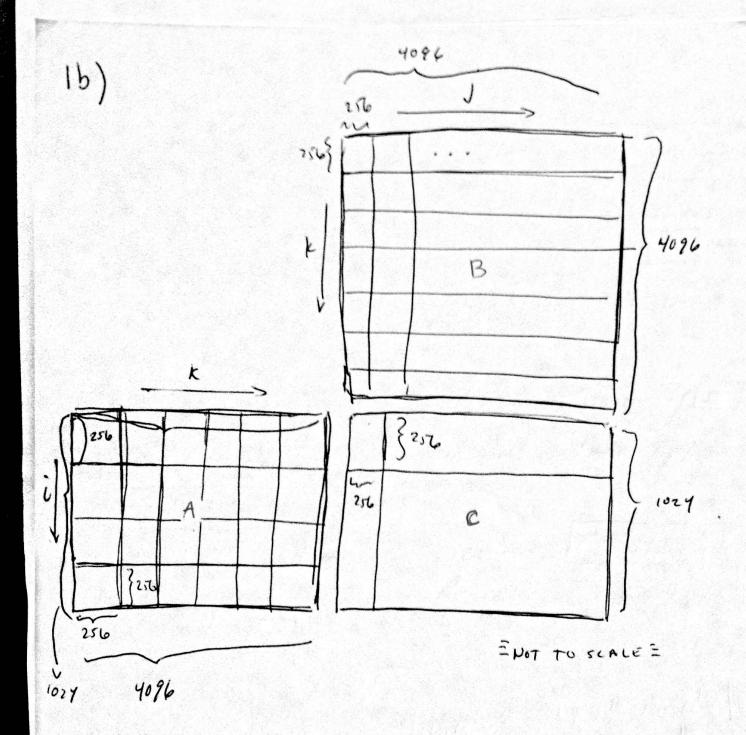


1) A will miss once per every fam addresses read:
1) A will miss once per every fact
4094/3 = 1024
il A is not indeped by i know all of the elector here is [1]
i) A is not independ by i know all of the elements of A accuracy in the to boy will fit in cacle memory, the factor here is [1] and the top will fit in cacle memory, the factor here is [1]
a de seu row we see new elements of A.
i) Each time on got a new row, we see new elements of A. i) Each time on got a new row, we see new elements of A. be each iteration will cause a miss. Ti = 5-12 he each iteration will cause a miss. Ti = 5-12 he each iteration will cause a miss. Ti = 5-12 he each iteration will cause a miss. Ti = 5-12 he each iteration will cause a miss. Ti = 5-12 he each iteration will cause a miss. Ti = 5-12
all A's indices, but as the inner loops will
to each iteration will cause a miss. It inner loops will get does not affect A's indices, but as the inner loops will get of a the cache can not affect addresses of A. The cache can not will cause
access 4096 x 5/2 a. Heren to Po cash iteration will care
get of our not effect A's indices, but as the mind cache can not access 1096 = 512 different addresses of A, The cache can not fold all of A that it must. No each iteration will cauce fuld all of A that it must. No each iteration will cauce
e miss, N; /1; 1701.
Il course us to switch
Sent time me mach a new block we will generate
Each time we mach a new stack we will generate a new sent each misses. N: /T: = 1024/512 = 2 Total = 210.20.21.23.21 =
9Total = 210.20.29.23.21 =
50 마이트 1997년 이번 1일
Explace iteration will take us to a new sow of B. Therefore we will miss each time. 4096
k) Each iteration with time. [409/0]
will miss each were brought
1) In the previous Heration 4086 x 4 elements were brought 1) In the previous Heration 4086 x 4 elements were brought 1) In the previous Heration 4086 x 4 elements were brought
to the cache. They all fit since 4016x4-1001
1) In the purious iteration MO86 x 4 elements were to be corrected to the cache. They all fit since 4096 x 4 = 1024 x 16 C1024 x 572 = 6. We will miss once for four boop iterations. To /4 = 512/4 = 27
i) By now we have brought a whole trie of B to the cache.
-) Bu now we have prought a wrote to from iteration is it
at a first by and all of the elements from iteration at i
I I woulded. We will miss he will miss he
i) By now we have brought a whole the of 8 to the cucke. If have the and all of the elements from iteration j= jt that been expelled. We will miss for every iteration of i.
T= 512

B (continued) B (continued) jt) We are now moving to a new tile. [1]: /T; = 4096 = 8 jt) We are now moving to a new tile. [N; /T; = 4096 = 8] the misses for each iteration
jt) We are now "
jt) We are now moving to a new tile. $N_i/T_i = \frac{7010}{512} = 8$ jt) We are now moving to a new tile. $N_i/T_i = \frac{7010}{512} = 8$ it) We will regulate all of the misses for each iteration of it. Then will be $N_i/T_i = \frac{1024}{512} = \frac{12}{2}$ such misses, Total = $\lambda^2 \cdot \lambda^3 \cdot \lambda^9 \cdot \lambda^3 \cdot \lambda^2 = \lambda^3 \cdot \lambda^4 \cdot \lambda^3 \cdot \lambda^4 \cdot \lambda^3 \cdot \lambda^4 $
of it. There will be Ni/Ti = 1/3/2 1/3/2 = 2/3/2 misses, Total = 2/2. 27. 29. 23.21 = 232
1) Vo misses are generated by this loop. Factor of 12/
j) Ewill be accessed by column, so there will be T; 13 =
1 1 1 - 107 mix CPS
i) Each iteraction will take us to a new row, $T_i = 872$ misses, it) Each iteraction will take us to a new row, $T_i = 872$ misses, jt) We will have more misses for each new tite.
$N_{j} / 2_{j} = 18 $
it) Same as for jt. [2] Total = 2°. 2+. 29. 25. 21 = [20]
do the total misses is [2234-237+220]



5,64
A K) A will miss the sache $T_{K}/B = 256/4 = 2^{5}/2^{2} = 2^{6}$ Lines i) A will only miss once. All of the k elements are in
i) A will miss the sache 1 x/8 i) A will only miss once . All of the k elements are in the cache after the first iteration. II the cache after the first iteration. Ti = 256
i) A will only miss once . All of the Har. [1]
the cache after the first row accessed. Ti = 25%
i) A will miss once for each new row accessed. Ti = 25%
11) A will miss each time we move to which
i) A will miss once to a new tile. At) A will miss each time we move to a new tile. NATE = 4096/256 = 16
to love enough to hold acl of
it) The cache isn't runge energy by the enner loops.
NATE = 4076/256 16 16 16 16 16 16 16 16 16 16 16 16 16
We will have accessed the
ANIPOSCO IVI - IVIO / C
it) Lant reason as gt we are accessing new rows of A with each iteration. Ni /Ti = 1024/256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 24 256 = 4 256
of A with each iteration. Ni /Ti = 1024/256
Total: 2".2".2".2".2".2"
<u>B</u> - <u>b</u> - <u>t</u>
K) Each iteration will go to a new row. In = [256] misses.
j) we will miss every fourth iteration, as the block size B=4.
t. 10 = 256/4 = 12 = 1011
i) bis not indested by i Furthermore, a whole the of
i) b is not indested by i Furthermore, a whole tie of B will lit in the cache. [1]

B Continued) N-17 = 4096/256 = 16
1. In will see a new pair of S. Z. Z.
it) The same thing happens. We are seeing a new part
it) All of B does not fit in the cache. 1024/256 = 4
it) All of B does not fit in the cacke. [1024/256 = 4] Stold = 28 26.20.24.24.22 C = 224 26.20.24.24.22
<u>c</u> = 2-1
K) lis not indeved by E. II
x) lis not indeved by 2. i) We will gotherough the columns of C. Tilb= 256/4 = [4]
256/4 = [69]
i) we will see a new row of C each Heration.
So there is a ractor of ti
At) A is not indexed by to. Also, a whole tile of
(At) C is not indexed by to. Also, a whole tile of C fits in the cache. Therefore, miss factor - F []
je) we will go to a new fait of - 1016/256 . [16]
(24) we are seeing new fairts of C.
1024/256 - [4] Notal = 2°, 2°, 28, 2°, 24, 22 = [520]
Words 20, 26, 28, 20, 20, 7 10-1

The total misses for loop a is
$$2^{25} + 2^{32} + 2^{20} = 2^{20}(3^5 + 2^{12} + 1)$$

= $2^{20}(8 + 4016 + 1)$
= $2^{20}(4105)$

The total misses for loop bis: $2^{24} + 2^{24} + 2^{20} = 2^{20} (16 + 16 + 1)$

So loop b is preferable.