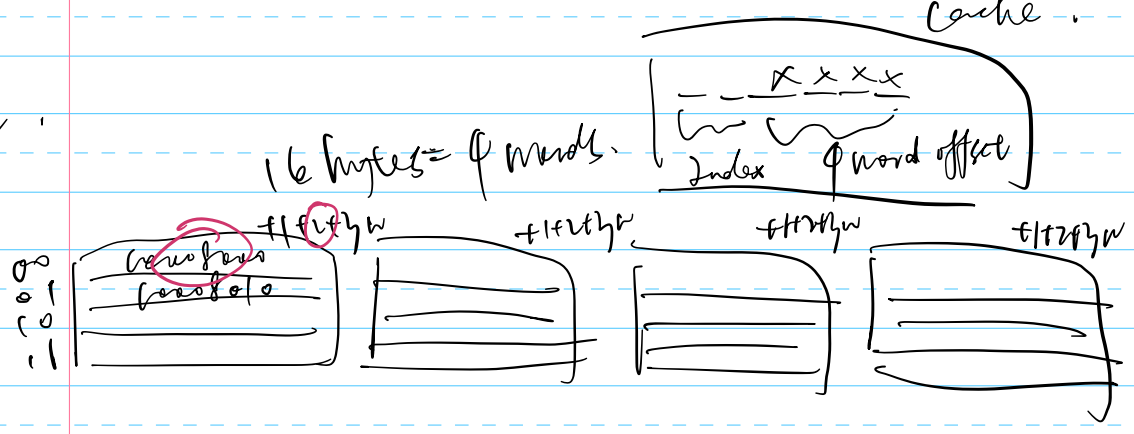
2. 0.

3. Swap inner and outer loops.

4.



5. 1 miss => 3 hits.

75%

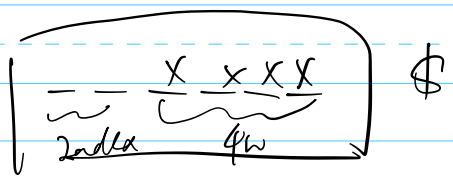
256 bytes.

= 64 w

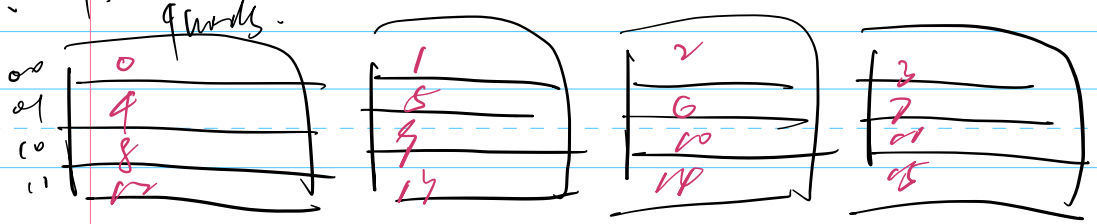
7. → 1 Am on 4.

4. Block_size = Cache size (256 Bytes)

```
for (j=0; j< array_size; j+= block_size){
    for (k=0; k< repeat; k++){
        for (i=0; i< block_size; i+=step){
            op(k, array[i+j]);
        }
    }
}
```



4-1. 1 words.



0.671875

0.59375

0.609375

0.578125

0.71875

1 miss => 1 hit

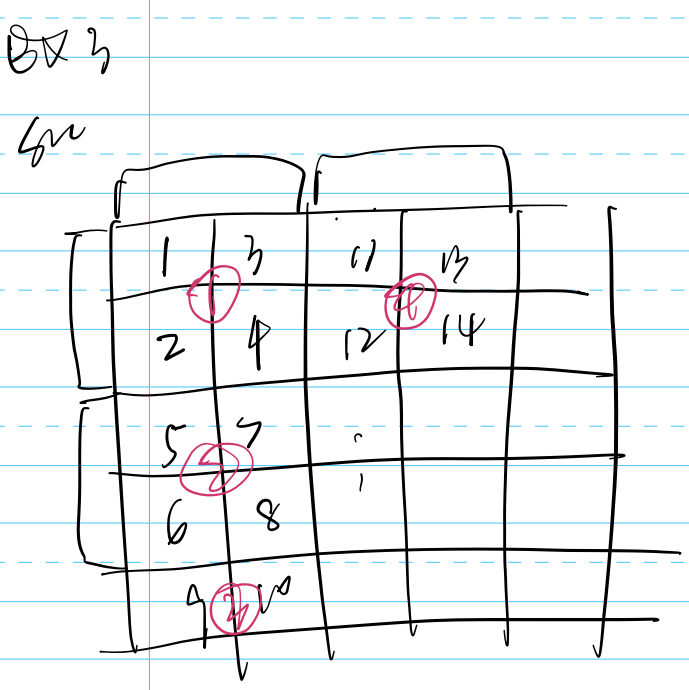
1 miss => 3 hit

0.5 ~ 0.75

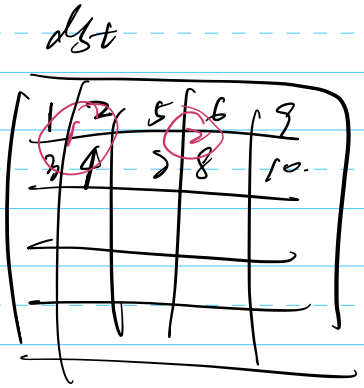
2. Random => LRU.

or Associativity: 4 => 1

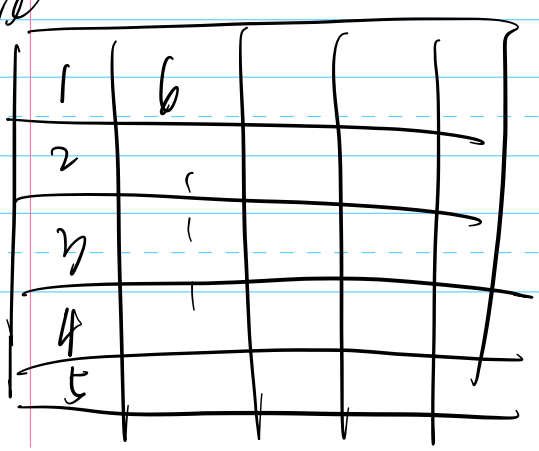
0.75 1 miss => 3 hits



n=5. b=2



naive



1.

1000	20	8.019	0.022
10000	20	42.15	0.867
20000	20	85.552	3.857
50000	20	104.412	26.872
60000	20	715.084	127.162

cache block offset term decay.

2.

10000	32	123.689	18.624
10000	1000		114.5
10000	5000		112.785
10000	10000		109.847
10000	50000		722.652

>> cache block offset