Discussion 10

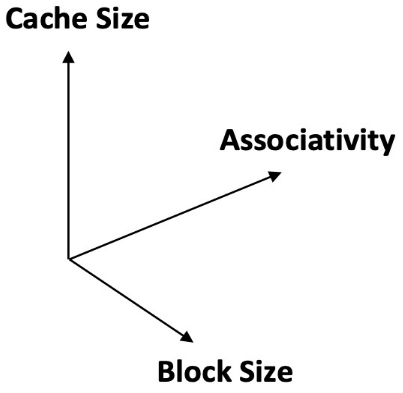
Cache

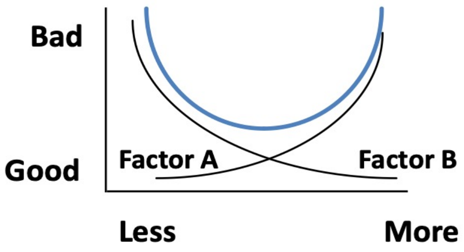
Zheng yuting

Review

In Conclusion, Cache Design Space

- Several interacting dimensions
  - Cache size
  - Block size
  - Associativity
  - Replacement policy
  - Write-through vs. write-back
  - Write-allocation
- Optimal choice is a compromise
  - Depends on access characteristics
    - Workload
    - Use (I-cache, D-cache)
  - Depends on technology / cost
- Simplicity often wins

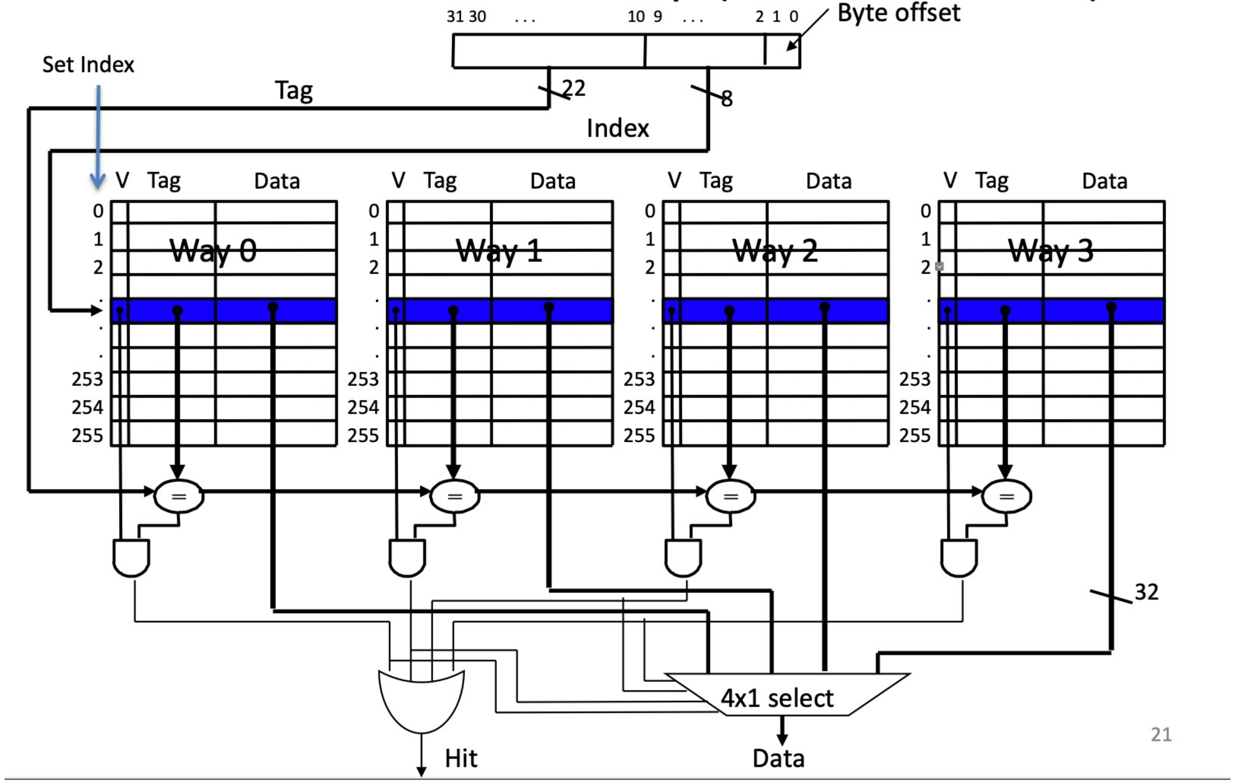




Review

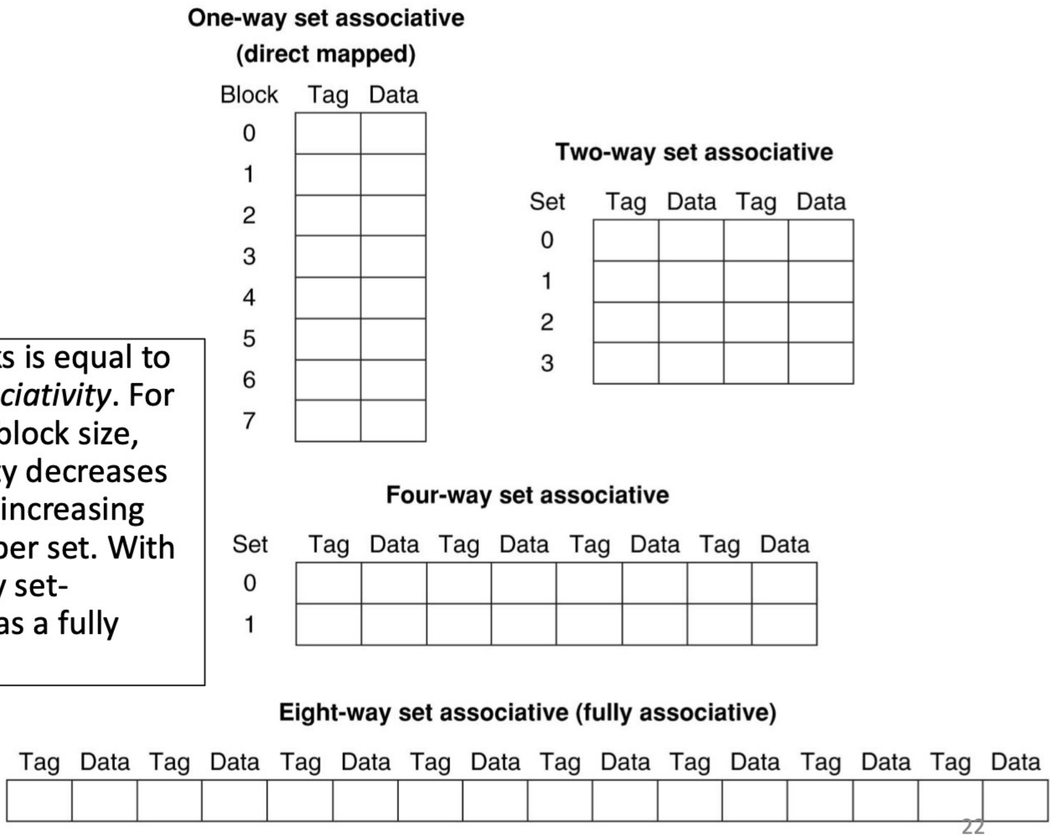
Four-Way Set-Associative Cache

- 2^8 = 256 sets each with four ways (each with one block)



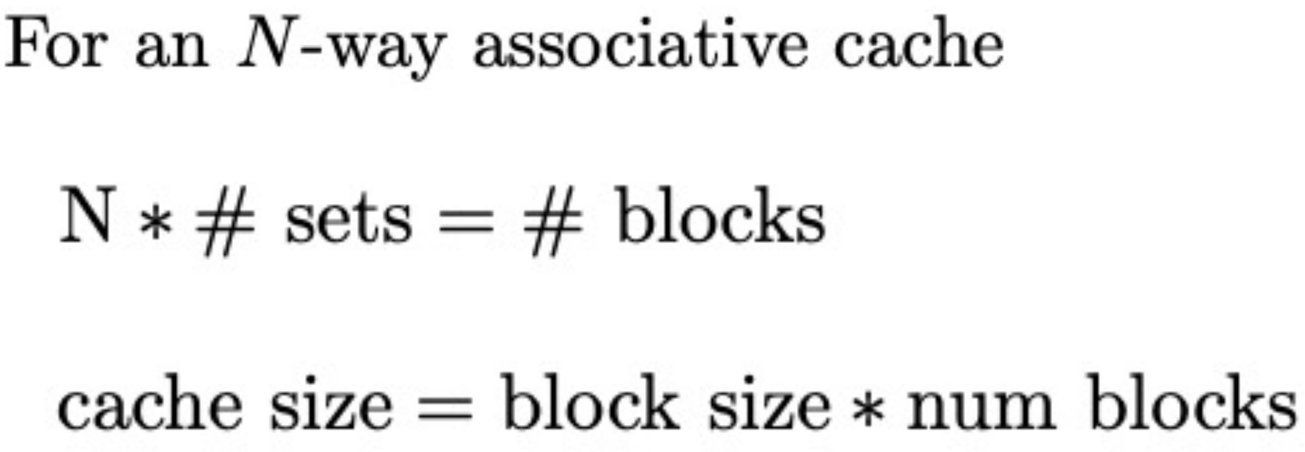
Review

Different Organizations of an Eight-Block Cache


One-way set associative
(direct mapped)
Block Tag Data
0
1
2
3
4
5
6
7
Two-way set associative
Set Tag Data Tag Data
0
1
2
3
Four-way set associative
Set Tag Data Tag Data Tag Data Tag Data
0
1
Eight-way set associative (fully associative)
Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data

Total size of $ in blocks is equal to *number of sets* × *associativity*. For fixed $ size and fixed block size, increasing associativity decreases number of sets while increasing number of elements per set. With eight blocks, an 8-way set-associative $ is same as a fully associative $.

Cache Calculation

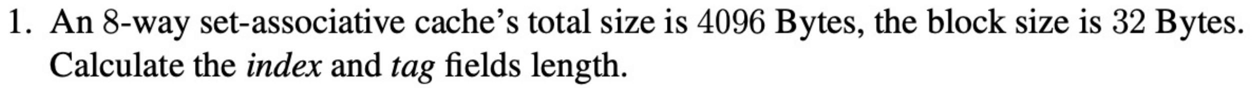


For an N-way associative cache

$$\mathrm{N} * \# \text{ sets} = \# \text{ blocks}$$

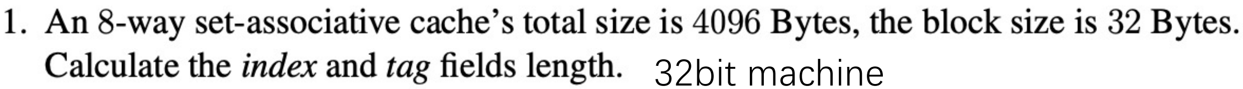
$$\text{cache size} = \text{block size} * \text{num blocks}$$

Cache Problem From Previous Exam



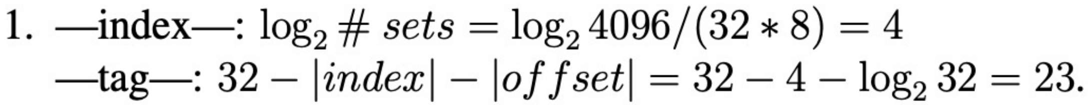
1. An 8-way set-associative cache's total size is 4096 Bytes, the block size is 32 Bytes. Calculate the *index* and *tag* fields length.

Cache Problem From Previous Exam



1. An 8-way set-associative cache's total size is 4096 Bytes, the block size is 32 Bytes. Calculate the *index* and *tag* fields length.

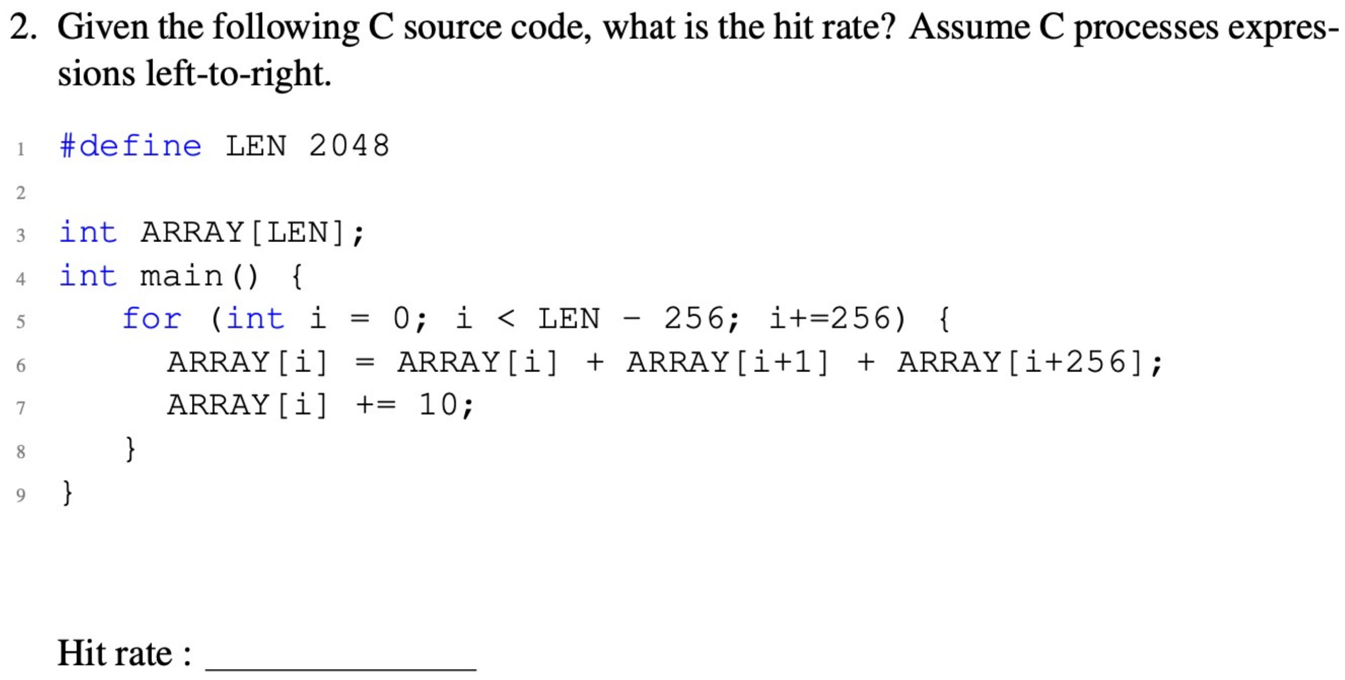
32bit machine



1. —**index**—: $\log_2 \# \, sets = \log_2 4096/(32 * 8) = 4$
   —**tag**—: $32 - |index| - |offset| = 32 - 4 - \log_2 32 = 23.$

Cache Problem From Previous Exam

TAG : 22, Set Index:6, Block offset: 4, Direct mapped cache



2. Given the following C source code, what is the hit rate? Assume C processes expressions left-to-right.

```
#define LEN 2048

int ARRAY[LEN];
int main() {
   for (int i = 0; i < LEN - 256; i+=256) {
     ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+256];
     ARRAY[i] += 10;
   }
}
```

Hit rate : _______________

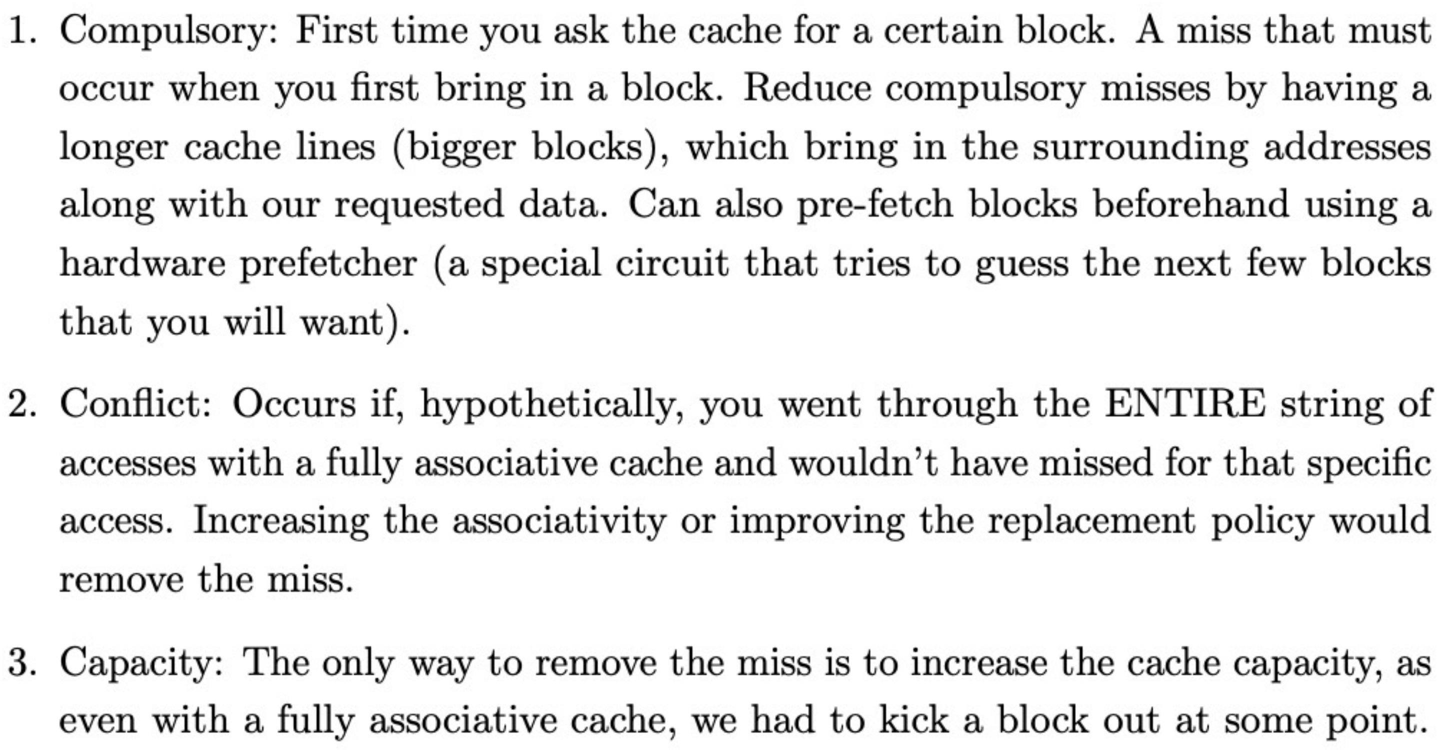
```c
#define LEN 2048

int ARRAY[LEN];
int main() {
   for (int i = 0; i < LEN - 256; i+=256) {
     ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+256];
     ARRAY[i] += 10;
   }
}
```

Hit rate : ________________

Solution: 50%
Every iteration it's
ARRAY[i] read MISS
ARRAY[i+1] read HIT
ARRAY[i+256] read MISS
ARRAY[i] write MISS (conflict! - same cache line as i+256!)
ARRAY[i] read HIT
ARRAY[i] write HIT
3 MISSES, 3 HITS. 50% hit rate.

3Cs

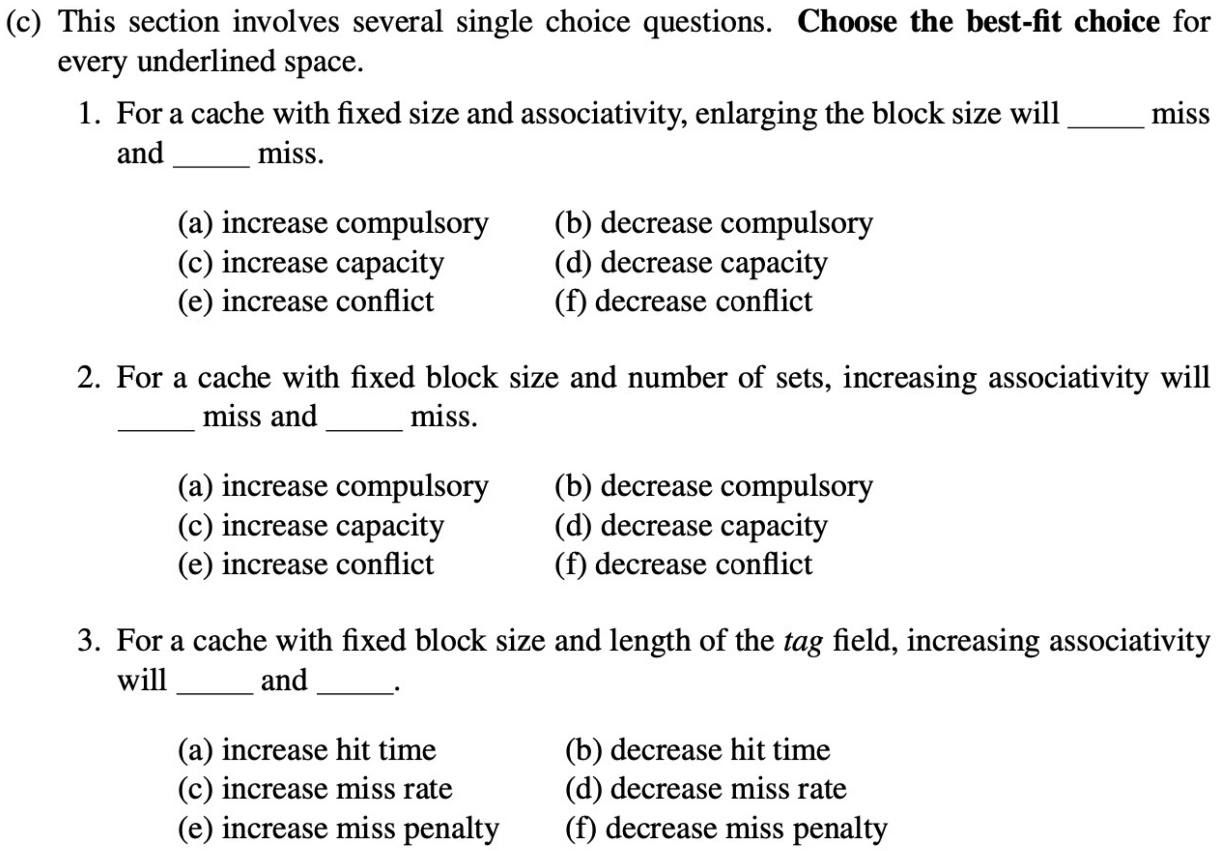


1. Compulsory: First time you ask the cache for a certain block. A miss that must occur when you first bring in a block. Reduce compulsory misses by having a longer cache lines (bigger blocks), which bring in the surrounding addresses along with our requested data. Can also pre-fetch blocks beforehand using a hardware prefetcher (a special circuit that tries to guess the next few blocks that you will want).

2. Conflict: Occurs if, hypothetically, you went through the ENTIRE string of accesses with a fully associative cache and wouldn't have missed for that specific access. Increasing the associativity or improving the replacement policy would remove the miss.

3. Capacity: The only way to remove the miss is to increase the cache capacity, as even with a fully associative cache, we had to kick a block out at some point.

Cache Problem From Previous Exam



(c) This section involves several single choice questions. **Choose the best-fit choice** for every underlined space.

1. For a cache with fixed size and associativity, enlarging the block size will _____ miss and _____ miss.

   (a) increase compulsory (b) decrease compulsory
   (c) increase capacity (d) decrease capacity
   (e) increase conflict (f) decrease conflict

2. For a cache with fixed block size and number of sets, increasing associativity will _____ miss and _____ miss.

   (a) increase compulsory (b) decrease compulsory
   (c) increase capacity (d) decrease capacity
   (e) increase conflict (f) decrease conflict

3. For a cache with fixed block size and length of the *tag* field, increasing associativity will _____ and _____.

   (a) increase hit time (b) decrease hit time
   (c) increase miss rate (d) decrease miss rate
   (e) increase miss penalty (f) decrease miss penalty

1. For a cache with fixed size and associativity, enlarging the block size will _____ miss and _____ miss.

   (a) increase compulsory (b) decrease compulsory
   (c) increase capacity (d) decrease capacity
   (e) increase conflict (f) decrease conflict

2. For a cache with fixed block size and number of sets, increasing associativity will _____ miss and _____ miss.

   (a) increase compulsory (b) decrease compulsory
   (c) increase capacity (d) decrease capacity
   (e) increase conflict (f) decrease conflict

3. For a cache with fixed block size and length of the *tag* field, increasing associativity will _____ and _____.

   (a) increase hit time (b) decrease hit time
   (c) increase miss rate (d) decrease miss rate
   (e) increase miss penalty (f) decrease miss penalty

Solution:

1. (b), (e) (The order doesn't matter)
2. (d), (f) (The order doesn't matter)
3. (a), (d) (The order doesn't matter)

Midterm Preparation

Instructor

Sören Schwertfeger

Sören Schwertfeger 师泽仁

<soerensch>

17

Instructor

Sören Schwertfeger 师泽仁

<soerensch>

18

Instructor

Sören Schwertfeger 师泽仁

<soerensch>

19-20

Q&A

THANKS FOR YOUR ATTENDANCE