CACHING STRATEGY

Huang Bo
itakejgo@gmail.com

CACHE STRATEGY

Using Caching can speed up our “I” only if we use efficient strategy…

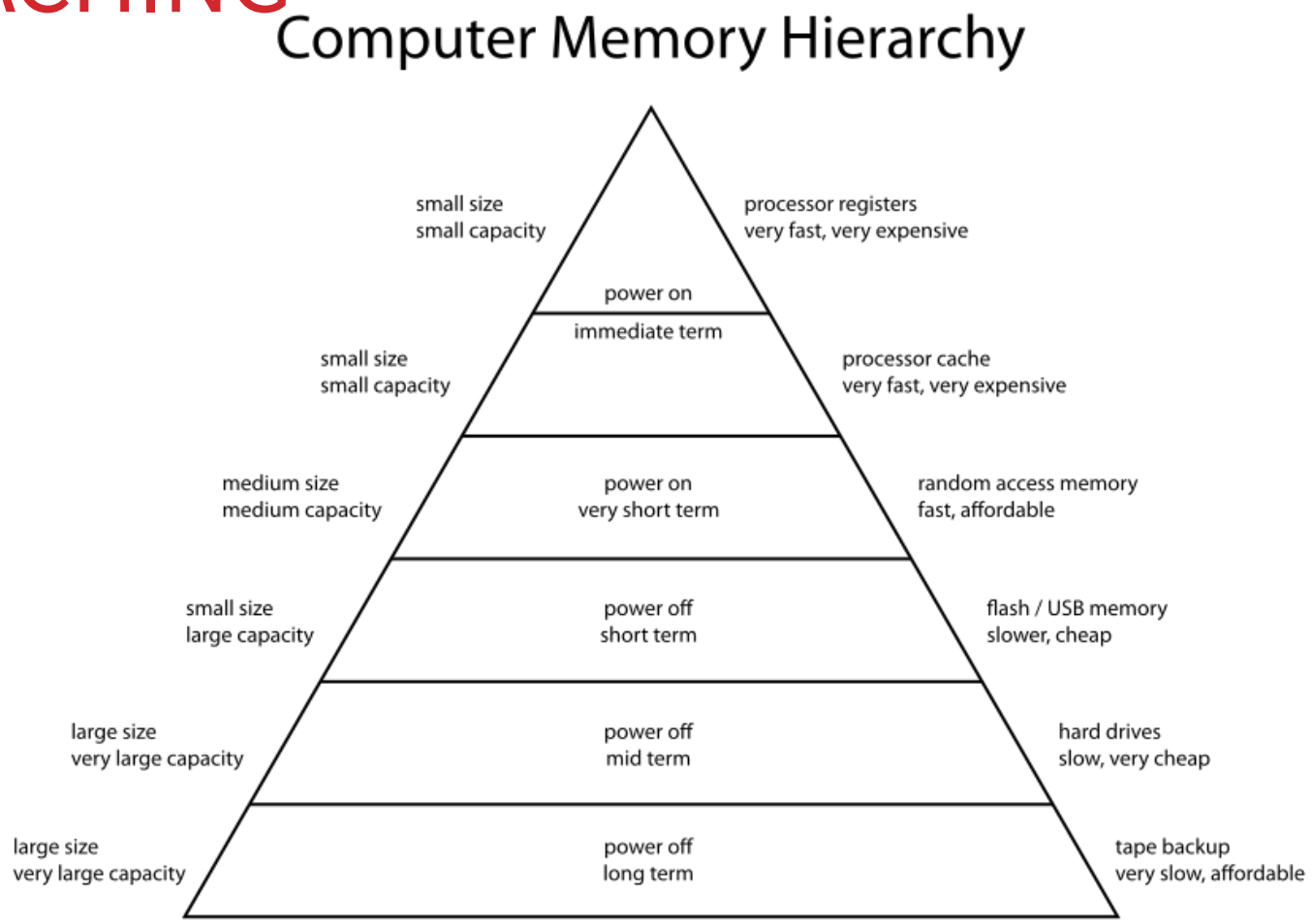
CACHE STRATEGY

Just like Time Complexity…

Traditional search in ordered sequence: O(n)

Binary search: O(lgn)

CACHING



Pictures from wiki

CACHE STRATEGY

If our miss percentage is 100%.

worse than no cache

If our miss percentage is 0%.

we don’t need to access main memory

CACHE STRATEGY

Cache storage —— input

Cache query —— output

CACHE STRATEGY

- **Direct mapped cache**
- **N-way set associative cache**
- **Fully associative cache**
- **…**
- **Etc.**

CACHE STRATEGY

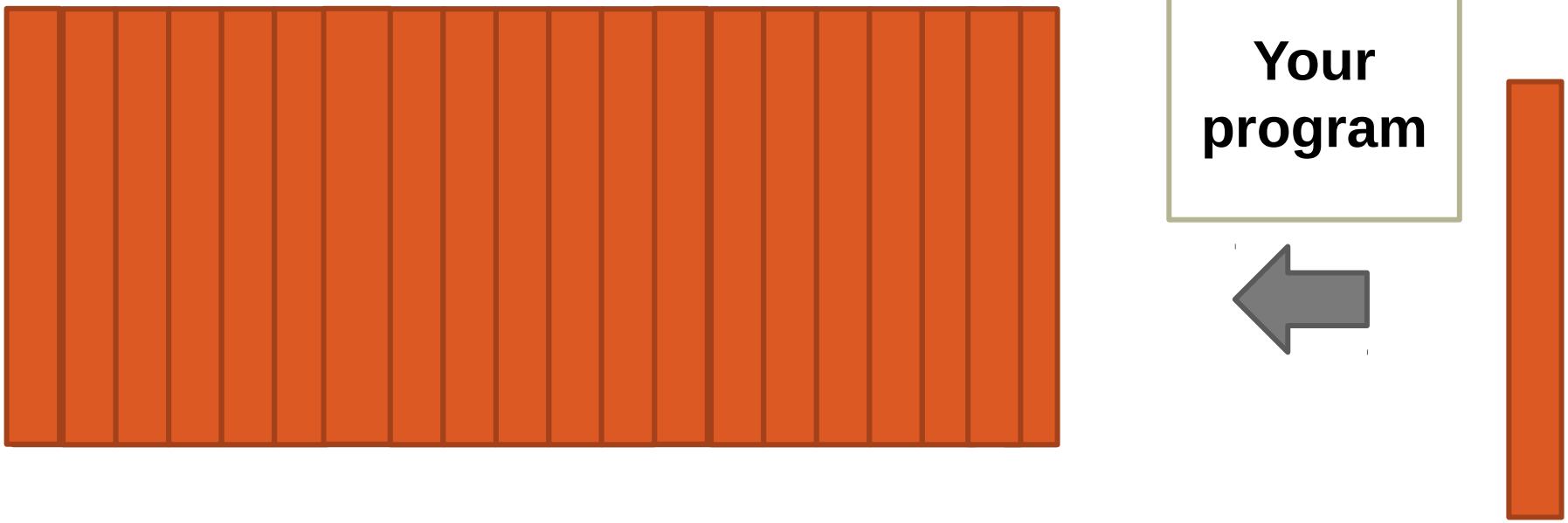
- **Direct mapped cache (our lab7)**
- **N-way set associative cache**
- **Fully associative cache**
- **…**
- **Etc.**

CACHE STORAGE

When cache is not full, we just do things as we wish.

CACHE STORAGE

When cache is full, which cache line should we replace ?



PAGE REPLACEMENT

FIFO

a queue or … ?

Min(Replace the one won’t be used for lon gest time)

too hard and why?

Random

if PRG axiom is true …

FIFO

Using a queue to maintain the page. So when there comes a new query:

1) If it exists, return the result. (Hit)

2) If it doesn’t exist. (Miss)

1. if the queue is full, pop

2. go to next level and …

FIFO

1) If it exists, return the result. (Hit)

2) If it doesn't exist. (Miss)

    1. if the queue is full, pop

    2. go to next level and …

FIFO

Q: Can we do better?

MIN

We replace the one won’t be used for longe st time.

Min gives us the minimum number of faults.

For each query,

1) If it exists, return the result. (Hit)

2) If it doesn’t exist. (Miss)

1. If the ? is full, find the one.

2. go to next level and …

MIN

1) If it exists, return the result. (Hit)

2) If it doesn't exist. (Miss)

   1. If the ? is full, find the one.

   2. go to next level and …

MIN

Q. Can you prove that Min algorithm gives t he minimum number of faults?

LRU

Replace the least recent used one.

For each query,

1) If it exists, return the result. (Hit)

2) If it doesn’t exist. (Miss)

1. If the ? is full, find the one.

2. go to next level and …

LRU

For each query,

1) If it exists, return the result. (Hit)

2) If it doesn't exist. (Miss)

1. If the ? is full, find the one.

2. go to next level and …

LRU

Q: Can we do better?

DEMO

I implement simple FIFO and LRU.

LAB REQUIREMENT

1. **Write or Complete the code, so that it can run FIFO, LRU, Min, Clock and second-chance algorithm.**
2. **Bonus: If you can improve and implement the time complexity of FIFO and LRU, you can get bonus point.**
3. **Package should be named as: OS_lab8_Name_xxxxxxxx where xxxxxxxx is your student id, Name is your name. This package should contain: your report, your code.**
4. **Check blackboard for ddl.**

THANKS