CACHE REPLACEMENT PROBLEM

LRU Algorithm

Zhiyuan Wang 12032878
Supervisor: Ke Tang

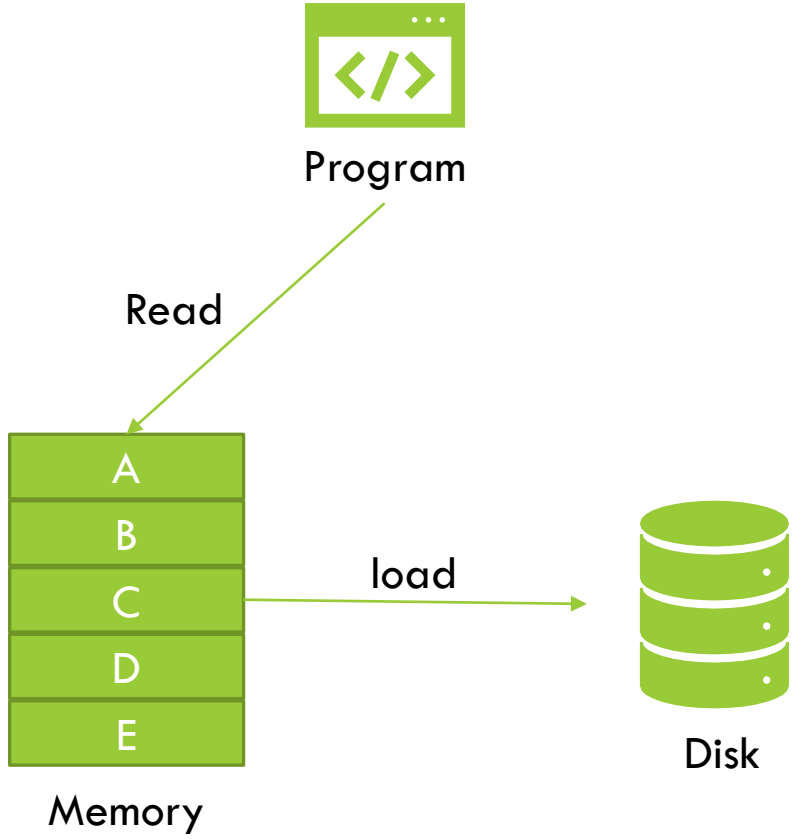
SUSTech Southern University of Science and Technology

CONTENT

1. Introduction to the Cache Replacement Problem.
2. Introduction to the LRU Algorithm

CACHE REPLACEMENT PROBLEM

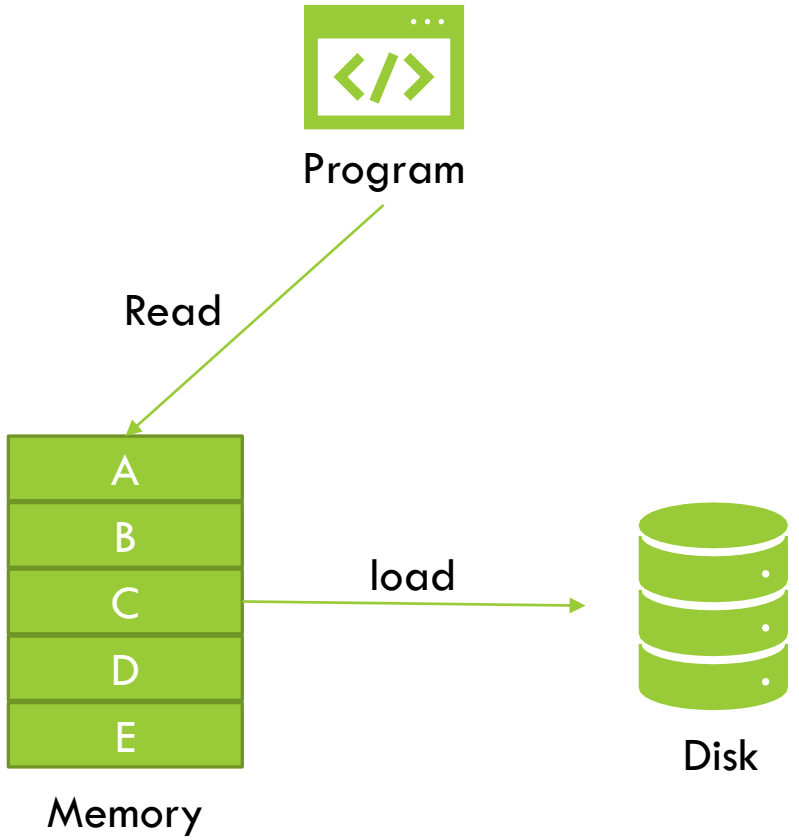
- INPUT:
  - A sequence of data accessing in the memory.
  - The capacity of the memory.
- OUTPUT:
  - The status of the memory.
- BACKGROUND:
  - It cost so much time to load data from disk.
  - It is quick to read data from memory directly.
  - Space of the memory is limited, can't save all the data.



- TARGET:
  - Scheduling the process to replace the data in the memory, minimize the times of loading data from disk.

LRU ALGORITHM

- LRU(Least Recently Used) Algorithm
  - This algorithm will delete the data that haven't be used for the longest time while the memory is full and there is new data which not in the memory and need to be loaded.
  - This algorithm think that the data we have used recently will be useful. Correspondingly, the data that we have used the least recently is the most impossible to be used soon, so it will be replaced.



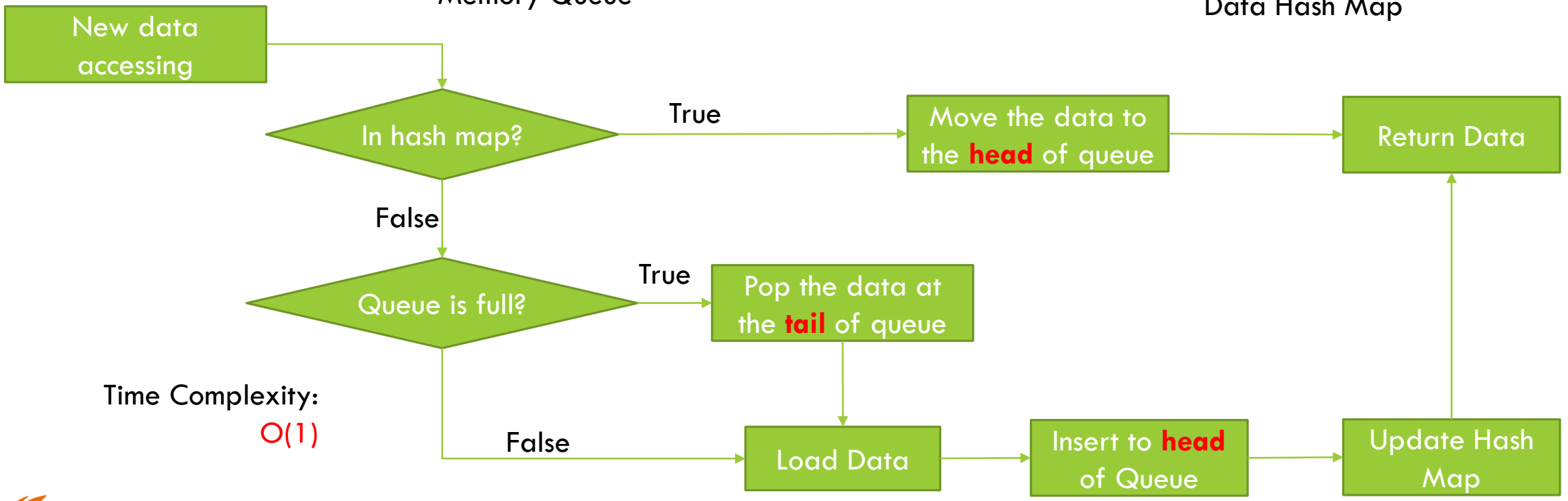
LRU ALGORITHM

Data 1	Data 2	.....	Data n

Memory Queue

Key	Value
Data Name	Data Address
.....	
Data Name	Data Address

Data Hash Map



Time Complexity:

O(1)

EXAMPLE OF LRU ALGORITHM

If the data sequence is ABCDEBEADC and the memory capacity is 5.
For the first 5 data accessing “ABCDE”, the memory is not full, so we can cache them directly.
The memory will be:

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

For the last 5 data accessing “BEADC”, the memory is full but all of them can be found in the hash map. So the we will move the data that is accessed to the head.

Next data item is B

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

For the last 5 data accessing “BEADC”, the memory is full but all of them can be found in the hash map. So the we will move the data that is accessed to the head.

Next data item is E

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

For the last 5 data accessing “BEADC”, the memory is full but all of them can be found in the hash map. So the we will move the data that is accessed to the head.

Next data item is A

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

For the last 5 data accessing “BEADC”, the memory is full but all of them can be found in the hash map. So the we will move the data that is accessed to the head.

Next data item is D

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

For the last 5 data accessing “BEADC”, the memory is full but all of them can be found in the hash map. So the we will move the data that is accessed to the head.

Next data item is C

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

After the sequence of data accessing “ABCDEBEADC”, If the next data item is F, the memory can’t store 6 items of data, so we need to replace an item using item F.

Memory Queue

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

After the sequence of data accessing “ABCDEBEADC”, If the next data item is F, the memory can’t store 6 items of data, so we need to replace an item using item F.

Memory Queue

We can find that B is the item that used the least recently, so we replace item B using item F.

Key	Value
A	A’s Address
B	B’s Address
C	C’s Address
D	D’s Address
E	E’s Address

Data Hash Map

EXAMPLE OF LRU ALGORITHM

After the sequence of data accessing “ABCDEBEADC”, If the next data item is F, the memory can’t store 6 items of data, so we need to replace an item using item F.

Memory Queue

We can find that B is the item that used the least recently, so we replace item B using item F.

Key	Value
A	A’s Address
C	C’s Address
D	D’s Address
E	E’s Address
F	F’s Address

Data Hash Map

THANK YOU !

Q&A

SUSTech Southern University of Science and Technology