Buffer Replacement Policies

- Consider the following query plan, chosen by the query optimizer, for implementing a join query between the borrower and customer relations (see pages 83 and 85 for schema descriptions).

```
select name, loan, street, city
from borrower, customer
where b[name] = c[name]
```

```
for each tuple $b$ of borrower do
    for each tuple $c$ of customer do
        if $b[customer-name] = c[customer-name]$ then begin
            $x[customer-name] := b[customer-name]$;
            $x[loan-number] := b[load-number]$;
            $x[customer-street] := c[customer-street]$
            include $x$ in the result of the query;
        end if;
    end for;
end for;
```
LRU Example

- Suppose that *borrower* consisted of blocks b1, b2, and b3, that *customer* consisted of blocks c1, c2, c3, c4, and c5, and that the buffer could fit five (5) blocks total.

- Applying the LRU replacement algorithm results in the following for the first two tuples of b1:

```
b1  b1  b1  b1  b1  b1  b1  b1  b1  b1
  c1  c1  c1  c1  c5  c5  c5  c4  c4
  c2  c2  c2  c2  c1  c1  c1  c1  c5  .......
  c3  c3  c3  c3  c2  c2  c2  c2  .......
  c4  c4  c4  c4  c3  c3  c3
```

- Notice that each time a customer block is read into the buffer, the block replaced is the next block required!

- This results in a total of six (6) page replacements.

- As an exercise, try an example using the MRU replacement algorithm.
A More Sophisticated Replacement Algorithm

- Read in a block of borrower, and keep it in the buffer until the last tuple in that block has been processed.

- Once all of the tuples from that block have been processed, free up the buffer space used by that block immediately (toss-immediately).

- If a customer block is to be brought into the buffer, and another block must be moved out in order to make room, then move out the most recently used block (borrower, or customer).

- Applying this replacement algorithm results in the following for the first two tuples of b1:

  b1  b1  b1  b1  b1  b1  b1
  c1  c1  c1  c1  c1  c1
  c2  c2  c2  c2  c2
  c3  c3  c3  c4
  c4  c5  c5

- This results in a total of two (2) page replacements.

- Note how using query-plan information results in a better buffer replacement algorithm.