Shiyi Li

ID: 4576123

CSCI 4707

Homework 3

A.

B.

1.

a. FIFO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | F | G | D | H |

b. MRU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | G | F | D | H |

c. LRU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | G | H | D | F |

d. Clock

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | G | H | D | F |

2.

DBMS can predict the order in which pages will be accessed, or page reference patterns, much more accurately than is typical in an OS environment.

DBMS needs more control over when a page is written to disk than an OS typically provides.

3.

a.

MRU, since the file will be scanned again and again, we can use most recent used policy to do that, first, we can place pages into buffer frame, and the put the next page into the first place, which is the most recent used, and the least recent used page will be at the last place.

b.

Clock sweep, in zipfian distribution, we know some pages will be requested frequently, and some of them will be requested rarely. So we better to keep the frequently used pages in the buffer, only replace the pages that rarely requested. The clock sweep is better fit on this situation because clock sweep replace the oldest unpin bit.

4.

a. When the qualifying records with respect equality search is small, it is better to use unflustered B+ tree. For unclustered B+ tree, each qualifying record will need one I/O to read the records from data. So it only needs a few I/Os since the qualifying records is small. We also need to read from root to leaf in the B+ tree, which also will take a few I/Os since the fan-out is large. For sorted file, it will need to use binary search to locate the desire record, and it takes longer than the unclustered B+ tree.

b.

No, because by the definition of alternative 1, which have to be clustered since it will cause the order of data record to be close to the order of entries.

c.

B+-tree only stores data in leaf nodes, and B-tree store data in interior nodes.

C

1.

a. sorted file scan

b. a hash index on state

c. file scan

d. clustered B+ tree index

2.

We can use clustered B+ tree index on <state, age> whith index only evaluation technique. We can use the B+ tree to pick out all the composite keys with state statisfying the state restriction for each state. Then we can calculate the average age for each state. In this case, we need I/O to read pages from root to first leaf, and then read through all the leaf pages. Since the index is sorted by state and age, so we can just calculate the average age by just reading through the index.

3.

a. 100+1000\*(2.5+1)=3600

b. 10+5000\*(1.2+1)=11010

c. 100+1000\*(2.5+5000/1000)=7600

d. 10+5000\*100= 500010