CS386D Database Systems
HW2 Solutions

8.4.1 (10 pts)

Action	No idx	Star idx	Movies idx	Both idx
Q1	100	4	100	4
Q2	100	100	4	4
I	2	4	4	6
Avg	$2 + 98p_1 + 98p_2$	$4 + 96p_2$	$4 + 96p_1$	$6 - 2p_1 - 2p_2$

8.4.2 (10pts)

Action	No idx	Name idx	Class idx	Launched idx	Name, Class idx	Name, Launched idx	Class, Launched idx	All idx
Q1	50	2	50	50	2	2	50	2
Q2	50	50	2	50	2	50	2	2
Q3	50	50	50	26	50	26	26	26
I	2	4	4	4	6	6	6	8
Avg	$2 + 48p_1 + 48p_2 + 48p_3$	$4 - 2p_1 + 46p_2 + 46p_3$	$4 - 2p_2 + 46p_1 + 46p_3$	$4 + 46p_1 + 46p_2 + 33p_3$	$6 - 4p_1 - 4p_2 + 44p_3$	$6 - 4p_1 + 44p_2 + 20p_3$	$6 + 44p_1 - 4p_2 + 20p_3$	$8 - 6p_1 - 6p_2 + 18p_3$

Eye-balling the average values, an index on Launched gives the least benefit. Index combinations on Name, Class should be considered against the cost of creating the indices. The final choice would depend on the relation between p1, p2, p3.

14.1.1 (10 pts)

a) Dense index: $\lceil n/3 \rceil + \lceil n/10 \rceil$ (first term for all data blocks, second term for all blocks pointing to each n)
b) Sparse index: $\lceil n/3 \rceil + \lceil n/30 \rceil$ (first term for all data blocks, second term for all blocks pointing to each block)

14.2.1 b. (5 pts)

A dense B+ tree must have 1000000 pointers to records in the leaves. Average B+ tree has 70 pointers.

1,000,000 /10 = 100000 record blocks
$\lceil 1000000/70 \rceil$ = 14,286 blocks for leaves
$\lceil 14286/70 \rceil$ = 205 blocks at next level
$\lceil 205/70 \rceil$ = 3 blocks at next level
1 block at root level
Total blocks = 114, 495

There are 4 levels from root to leaf => Average disk I/Os to retrieve a record is 5

14.2.1 c. (5 pts)

A sparse B+ tree must have 1 pointer from the leaf er block.

1,000,000 /10 = 100000 record blocks
$\lceil 100000/70 \rceil$ = 1429 blocks for leaves
$\lceil 1429/70 \rceil$ = 21 blocks at next level
1 block at root level
Total blocks = 101, 451

There are 3 levels from root to leaf => Average disk I/Os to retrieve a record is 4

Note: if you think leaf node contains 69 pointers instead of 70 pointers (one pointer pointing to the next leaf node), the answer for 14.2.1 b) should be 114705 blocks and 14.2.1 c) should be 101472 blocks. Those two are also correct.

14.2.2 b. (5 pts)

Total number of blocks remains the same.

The traversal of the index starts with 4 I/Os to locate the first leaf node.
Once the first leaf node is retrieved, we need $\lceil 1000/70 \rceil - 1 = 14$ I/Os to retrieve the rest.
Records are not sorted so 1000 records can be spread over 1000 blocks. This would take 1000 I/Os additionally.
Total I/Os = 4+14+1000 = 1018

14.2.2 c. (5 pts)

Total number of blocks remains the same.

The traversal of the index starts with 3 I/Os to locate the first leaf node.
Once the first leaf node is retrieved, we need $\lceil 100/70 \rceil - 1 = 1$ I/Os to retrieve the rest.
Records are sorted so 1000 records are on 1000/10 = 100 blocks. This would take 100 I/Os additionally.
Total I/Os = 3+1+100 = 104