1. Average seek time = 10ms
Average rotational latency = 5ms
Transfer time = 1ms for 4KB
MM = 320 blocks
# of read/write blocks = 10,000,000
320 * 4KB = 1.28

Phase 1: $\frac{10{,}000{,}000}{320}$ = 31,250 sorted runs

Phase 2: Merge: 319-way merges, so

$\frac{31{,}250}{319}$ = 98 sorted sublists

Phase 3: 2PMMS for the total # of I/O's

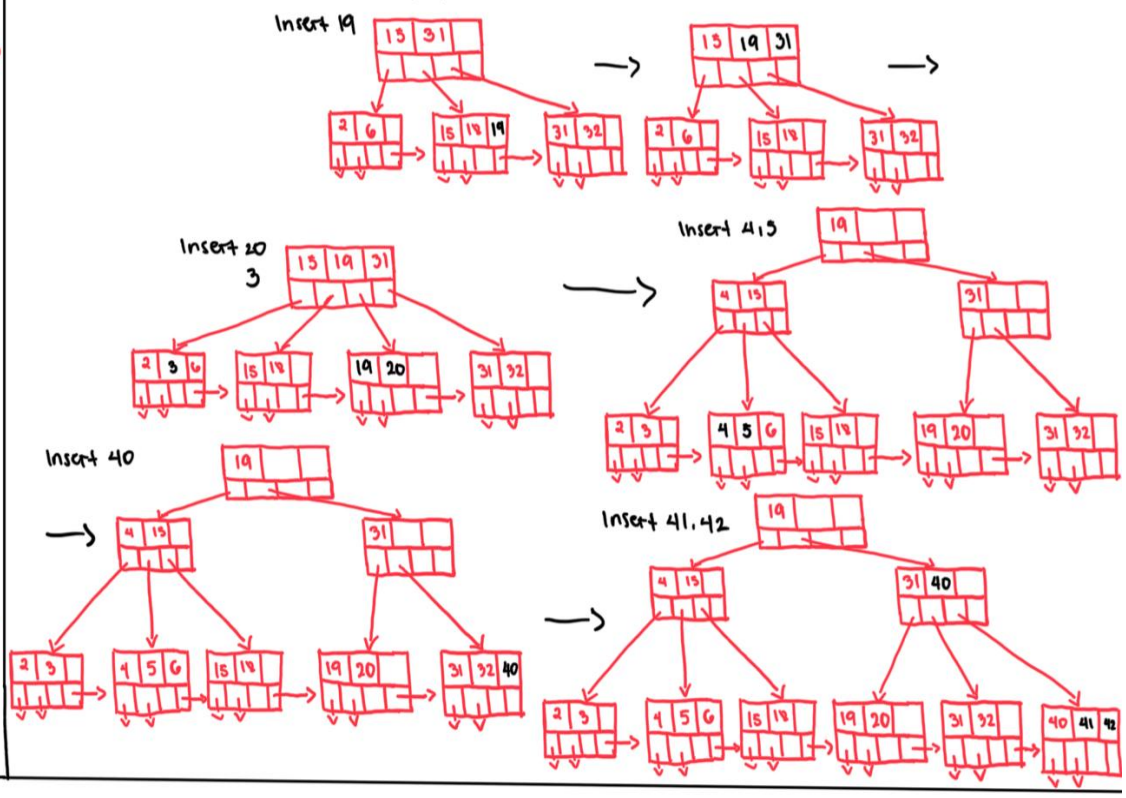
$3 \times 2 \times 10^7 = 6 \times 10^7$

too big

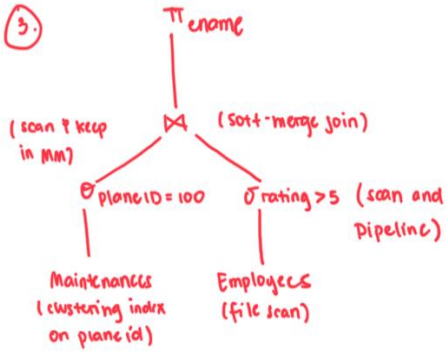
Total: $6 \times 10^7 \times 16ms$ = 960000000 ms or 11 days

2. n = 3 keys

~~2, 15, 31, 32, 6, 15, 19, 20, 3, 4, 5, 40, 41, 42~~



3.



Pushing selection II: cost of applying $\sigma_{planeId=100}$

Maintenance = 1000 blocks / 100 = 10

PlaneId clustering = 2 I/Os + 10 I/Os = 12

Pushing selection III: cost of applying rating > 5

50 + 25 = 75 I/Os

for scan and pipeline & MM

Pushing selection IV: sort-merge join

2PMMS cost of sorting

2 * 2 * 10 = 40

10 + 25 = 35

maintenance employees

12 + 75 + 40 + 35 = 162 I/Os

④ T_1, T_2, T_3

1. $r_1(A)\ r_2(B)\ r_3(C)\ r_1(B)\ r_2(C)\ r_3(D)\ w_1(A)\ w_2(B)\ w_3(C)$

i.

T_1	T_2	T_3
$xl_1(A); r_1(A)$		
	$xl_2(B); r_2(B)$	
		$xl_3(C); r_3(C)$
$sl_1(B)$; denied		
	$sl_2(C)$; denied	
		$sl_3(D); r_3(D)$
		$W_3(C); U_3(C); U_3(D);$
	$sl_2(C); r_2(C)$	
	$W_2(B); U_2(B); U_2(C);$	
$sl_1(B); r_1(B)$		
$W_1(A); U_1(A); U_1(B)$		

The first three reads were granted, and the next two requests were denied because both T_1 and T_2 aren't granted the shared lock due to the exclusive locks on B & C respectively. T_3, however, is granted a shared lock, so it releases the locks on C & D. From there, T_2 & T_1 gets a shared lock for some X and releases the lock.

ii.

T_1	T_2	T_3
$sl_1(A); r_1(A)$		
	$sl_2(B); r_2(B)$	
		$sl_3(C); r_3(C)$
$sl_1(B); r_1(B)$		
	$sl_2(C); r_2(C)$	
		$sl_3(D); r_3(D)$
$xl_1(A); w_1(A); U_1(A);$ $U_1(B)$		
	$xl_2(B); w_2(B); U_2(B); U_2(C);$	
		$xl_3(C); w_3(C); U_3(C); U_3(D);$

The first 6 read actions and shared locks were granted. No requests are denied by the scheduler and update locks were upgraded to exclusive locks.

iii.

T_1	T_2	T_3
$ul_1(A); r_1(A);$		
	$ul_2(B); r_2(B);$	
		$ul_3(C); r_3(C);$
$sl_1(B)$; denied		
	$sl_2(C)$; denied	
		$sl_3(D); r_3(D);$
		$W_3(C); U_3(C); U_3(D)$
	$sl_2(C); r_2(C);$	
	$W_2(B); U_2(B); U_2(C);$	
$sl_1(B); r_1(B);$		
$W_1(A); U_1(A); U_1(B)$		

The first three reads were granted. The request for a shared lock on T_1 & T_2 were denied because T_2 & T_3 already has an update lock. A scheduler will not grant a shared lock when there is an update lock.

2. $r_1(A); r_2(B); r_3(C); r_1(B); r_2(C); r_3(A); w_1(A); w_2(B); w_3(C);$

i.

T_1	T_2	T_3
$xl_1(A); r_1(A)$		
	$xl_2(B); r_2(B)$	
		$xl_3(C); r_3(C)$
$sl_1(B)$; denied		
	$sl_2(C)$; denied	
		$sl_3(D)$; denied

The first three read actions and exclusive locks were granted. But, requests for a shared lock results to a deadlock & no transaction proceeds. this is because the requests for some $sl_i(x)$ already has an exclusive lock for some transaction T_i

ii.

T_1	T_2	T_3
$sl_1(A); r_1(A)$		
	$sl_2(B); r_2(B)$	
		$sl_3(C); r_3(C)$
$sl_1(B); r_1(B)$		
	$sl_2(C); r_2(C)$	
		$sl_3(D); r_3(D)$
$xl_1(A)$; denied		
	$xl_2(B)$; denied	
		$xl_3(C)$; denied

The first 6 shared locks and read actions were granted. The next requests were denied and resulted to a deadlock. The request for an upgrade to exclusive is denied by the scheduler because the request for some lock X already has a shared lock for some transaction T_i

iii.

T_1	T_2	T_3
$ul_1(A); r_1(A);$		
	$ul_2(B); r_2(B);$	
		$ul_3(C); r_3(C);$
$sl_1(B)$; denied		
	$sl_2(C)$; denied	
		$sl_3(A)$ denied

The first three read actions were granted, but the next requests resulted to a deadlock. This is because request for some $sl_i(x)$ gets denied due to an update lock on the same X for some transaction T_i

ex. T_2 is granted a shared lock on C that then unlocks B & C. T_1 is then granted with a lock on B that then unlocks A & B.

5.

a.) R(A, B, C, D)

AB → C

B → D violation!

CD → A

AD → B

$R_1(B, D)$, $R_2(B, A, C)$

A+ = A

B+ = D, B

C+ = C

D+ = D

AB+ = A, B, C, D

BC+ = B, C, D, A

CD+ = C, D, A, B

AD+ = A, D, B, C

ABCD

AD = BC

AB = CD

CD = AB

b.) No BCNF violations

A → B

B → C

C → D

D → A

R(A, B, C, D)