Assignment 8
Prudhvi Vajja

Question 6

a) Specify (in ms) the minimum time to retrieve a record with key k in the B$^+$-tree provided that there is a record with this key.

Ans:

The largest integer n:
N <= block size – ((block-address size)/ (block-address size + record key size))
Extra block access id: N + 1

N = 4096 – 9 / (9+12)
N = 194
New N = 194 + 1= 195.

Given block access time as 10ms
Minimum time to retrieve a record is: $(\log_{195}(10^8) + 1) * 10 = 50$ ms.

b) Specify (in ms) the maximum time to retrieve a record with key *k* in the B+-tree.

Ans:

As N = 194 therefore for a branching factor of 2, no: of branches=> 194/2 + 1 = 94
The height of the tree is 4.

Block access time is 10ms
The total time is (4 + 1 + 1) *10 ms = 60 ms.

Question 9

a) Ans:

Records in R = 1,500,000.
Records in S = 5,000.
b(R) = 1500000/30 = 50000
b(S) = 5000/10 = 500
buffer blocks = 101

With R as outer relation:
IO = b(R) + (b(R) * b(S))/100 = 300000

With S as outer relation:
IO = b(S) + (b(R) * b(S))/100 = 250500

b) Ans:

Merge sorting R takes 2*b(R) $\log_{100}$(b(R)) = 2*50000 * log (50000) = 300000

Merge sorting S takes 2*b(S) $\log_{100}$(b(S)) = 2*500 * log (500) = 2000

Total = b(R) + b(S) + 2*b(R) $\log_{100}$(b(R)) + 2*b(S) $\log_{100}$(b(S))
= 352500

c) Ans:

Merge sorting R takes 2*B(R) $\log_{100}$(B(R)) = 2*50000 * log (50000) = 300000
Merge sorting S takes 2*B(S) $\log_{100}$(B(S)) = 2*500 * log (500) = 2000

With 1 B value i.e p=1:
we will do a block nested - loop join with s as outer relation.
So, it will require b(S) + b(R)b(S)/100 block accesses i.e 250500 block accesses.

With 2 B value i.e p=2:
we will do a 2-block nested - loop join with s as outer relation.
b(S) = 500/2 = 250, b(R) = 50000/2 = 25000.
So, it will require b(S) + b(R)b(S)/100 block accesses i.e 62750 block accesses.

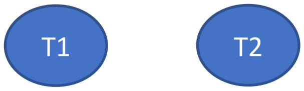
With 3 B value i.e p=3:
we will do a 3-block nested - loop join with s as outer relation.
b(S) = 500/3 = 170, b(R) = 50000/3 = 17000 nearly.
So, it will require b(S) + b(R)b(S)/100 block accesses i.e 29070 block accesses.

d) Ans:

Total cost
–Hash phase costs2*b(R)+2*b(S)
–Merge phase costs b(R) +b(S)
–Total: 3*(b(R)+b(S)) = 3*(50000+ 50) = 150150.

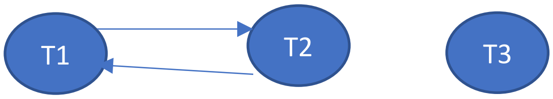
Part 3
Question 10

(a) S1 = R1(x)R2(y)R1(z)R2(x)R1(y)



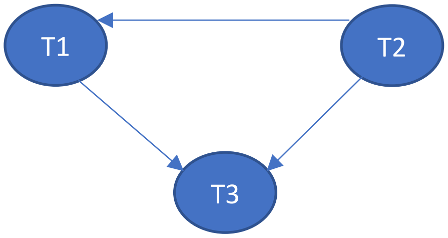
As there are no cycles, it is conflict serializable,
Conflict equivalent schedule: R1(x) R1(z) R1(y) R2(x) R2(y)

(b) S2 = R1(x)W2(y)R1(z)R3(z)W2(x)R1(y).



It is not conflict serializable, as there are cycles.

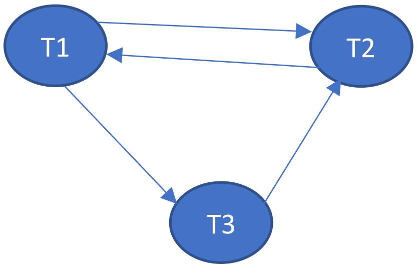
(c) S3 = R1(z)W2(x)R2(z)R2(y)W1(x)W3(z)W1(y)R3(x).



As there are no cycles, it is conflict serializable,
Conflict equivalent schedule: R1(z) W2(x) R2(y) R2(z) W1(x) W3(z) W1(y) R3(x)

Question 11

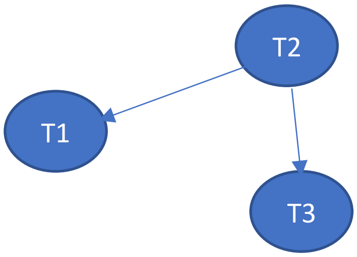
Give 3 transactions T1, T2, T3 and a schedule S on these transactions whose precedence graph (i.e. serialization graph) consists of the edges (T1, T2), (T2, T1), (T1, T3), (T3, T2).



R1(A) R2(B) W3(A) W1(B) W2(A)

Question 12

Give 3 transactions T1, T2, and T3 that each involve read and write operations and a schedule S that is conflict-equivalent with all serial schedules over T1, T2, and T3.



This is represented as: R2(x) R1(x) W1(x) R2(y) W2(y) R3(y) W3(y)

As T2 should execute before T1 and T3, so the order of T3 and T1 is not important Therefore the possible conflict equivalent schedules are:

1. T2 T1 T3

   T1 = R1(x)W1(x), T2 = R2(x) R2(y)W2(y), T3 = R3(y) W3(y)

2. T2 T3 T1

   T1 = R1(x)W1(x), T2 = R2(x) R2(y)W2(y), T3 = R3(y) W3(y)

As we have three Transactions: T1, T2, T3 Possible combinations are 6 out which four of them except this two have conflicts such as T1 T3 T2, T1 T2 T3, T3 T1 T2, T3 T2 T1

Question 13

a) **Show that each serial schedule involving transaction T_1 and T_2 pre- serves the consistency requirement of the database.**

For T1 T2:
Initially: A = 0, B= 0
After T1: A = 0, B = 1
After T2: A = 0, B = 1
A = 0 ∨ B = 0 ≡ T ∨ F = T

Similarly, for T2 T1 also consistency is met.

b) **Construct a schedule on T1 and T2 that produces a non-serializable schedule.**

R1(A); R2(B); R2(A); R1(B);
if A = 0 then B: = B+1;
if B = 0 then A: = A+1;
W2(A); W1(B);

c) **Is there a non-serial schedule on T1 and T2 that produces a serializable schedule? If so, give an example**

No, As R1(A) conflicts with W2(A) and R2(B) conflicts with W1(B). We cannot start with either of them without forming a cycle.