Question 1

1. Using a binary search, we need to divide the search range by two, and see whether the value we are searching is greater or less than the middle value of our split, we keep doing that until we find the value. While using an index, we can point out directly towards the right direction.

2. Yes , it is possible to reduce the the global depth by 2 through deletion. According to Zhang, Manolopoulos, Theodoridis, and Tsotras (2009), deletion may cause a bucket to become empty, and merge to it's buddy bucket. Moreover, many deletions can also halve its size, therefore reducing the global depth (Zhang, et al., 2009).

   Example:

0000	8*
….	
1000	4*
….	
1111	

   In order to distinguish the data entry 8* and 4*, the global and local depth needed for bucket 0000 and 1000 are 4. When we delete 4*, we can merge the buckets. After deleting 4*, the bucket 1000 is empty, therefore 0000 and 1000 can be merged into 000, reducing both local and global depth to 3. Furthermore, bucket 000 and 100 can be merged into 00.

00	8*
10	

   So we can see that global depth is decreased from 4 to 2.

3. When storing large volume of records, and we need to split in every single area, and not just in the dense area.

Question 2

-- Please check the appendix

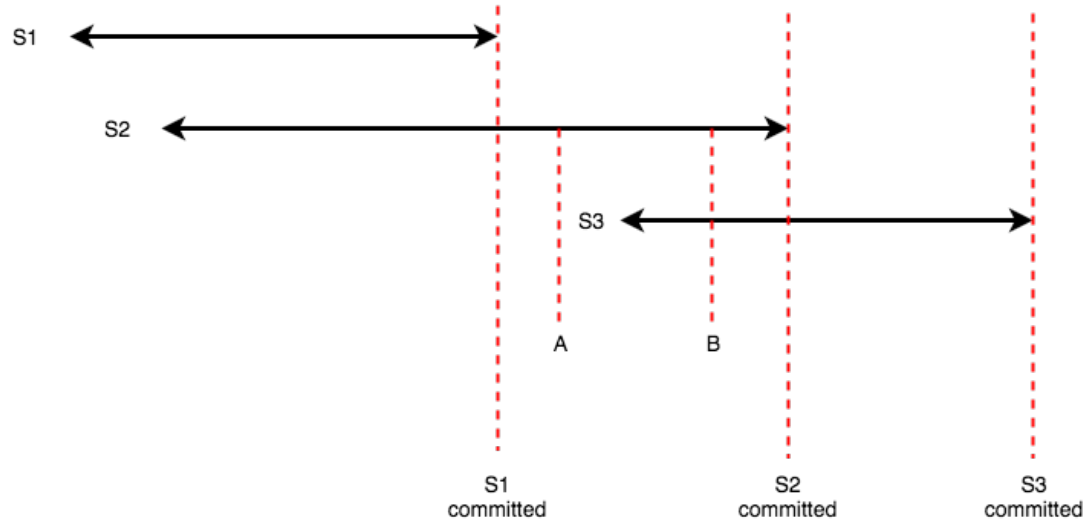
Question 3

With the assumption provided, it is believed that using a clustered B+ tree index on attributes R.a, R.b is the best method. This is due to the great capability of B+ tree in doing range search. Therefore, it can be said that the first data entry that

fulfils the condition can be discovered within log n time, then scan the rest of the data entry to obtain data of R.a and R.b. Since data entry is smaller than data record, more data can be retrieved in a page, which also reduces the IO cost, as opposed to acquiring data from data record.

Question 4

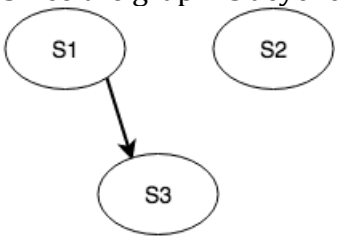
S1, R1(X), S2, R2(Y), W1(X), E1, A, R2(X), S3, R3(X), B, W2(Y), E2, R3(Y), W3(X), W3(Z), E3



1. Utilise the system log to recover the information that has been lost during the crash (Log-based recovery).

2. We need to redo everything starting from S2.

3. Since the graph is acyclic, then it is a serialisable schedule



S1 → S3

References

D.Zhang, Y.Manolopuolos, Y.Theodoridis, and V.J. Tsotras, 'Extendible Hashing', *Encyclopedia of Database Systems*, pages 1093-1095, L. Liu and M. Tamer Ozsu (editors), Springer, 2009.

Question 2.1