CSE370 : Database Systems
Practice Sheet
Indexing and Hashing

*Solutions are at the end of the document. SOLVE IT YOURSELF first.

Question 1:

An EMPLOYEE file with ID# as the key field includes records with the following ID# values: 8, 5, 1, 7, 3, 12, 9, 6. Suppose that the search field values are inserted in the given order in a B+-tree of order n = 3; show how the tree will expand and what the final tree will look like.

Question 2:

An EMPLOYEE file with ID# as the key field includes records with the following ID# values: 8, 5, 1, 7, 3, 12, 9, 6. Suppose that the search field values are inserted in the given order in a B+-tree of order n = 4; show how the tree will expand and what the final tree will look like.

Question 3:

A. An EMPLOYEE file with ID# as the key field includes records with the following ID# values: 15, 5, 20, 10, 65, 55, 60, 50, 30, 25, 35, 90, 85, 95, 105, 100, 110, 115. Suppose that the search field values are inserted in the given order in a B+-tree of order n = 4; show how the tree will expand and what the final tree will look like.

B. After building the tree in this sequence, describe in your words how you will search for these values in the tree: 65 and 120.

Question 4:

A. Discuss the similarities and differences between indexing and hashing.
B. Compare and contrast primary and secondary indexing with examples
C. What is multilevel indexing? Discuss the necessity of multilevel indexing.
D. Compare and contrast sparse indexing and dense indexing.

Question 5:

Suppose you have a table titled “‘Instructor”.

10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

A. Make a hash index on attribute instructor_id. Suppose the number of buckets in the hash index is 5 and each bucket can house 2 index entries at max. Bucket overflow is handled using forward chaining.

The hash function, h = (sum of all digits in id) % 5

B. Explain how to search for the following search keys in your hash index.
   - 45565
   - 11113

Question 6:

Suppose you have a table titled “‘Instructor”.

10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Make a hash index on attribute instructor_name (2nd column). Suppose the number of buckets in the hash index is 5 and each bucket can house 3 index entries at max. Bucket overflow is handled using forward chaining. The values of each character from A-Z (or a-z) is 1-26 respectively

The hash function, h = (sum of the values of each character in name) % 5

Ex:

h(Wu) = (23 + 21) % 5 = 4

Question 7:

Suppose you have the table below for “Instructors” for a School Database with columns ID, Name, Subject and Salary respectively:

76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
33465	Gold	Physics	87000

Suppose the number of buckets in the hash index is 5 and each bucket can hold 3 index entries at max. Construct a hash index on attribute ID. Bucket overflow is handled using forward chaining. The hash function, h = (First digit of ID) % 5.

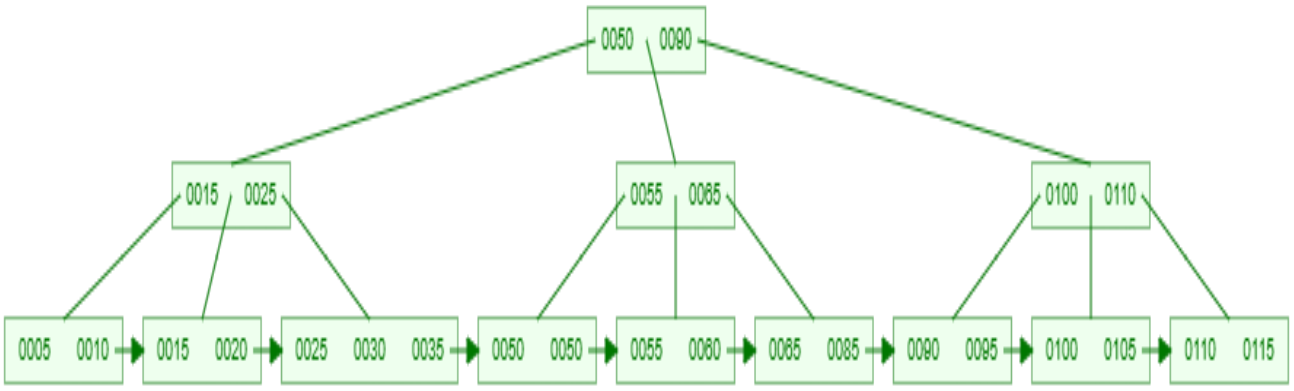
Now let's assume there is another hash function, h = (Last digit of ID) % 5.

Now compare which hash function is more uniform.

Question 8:

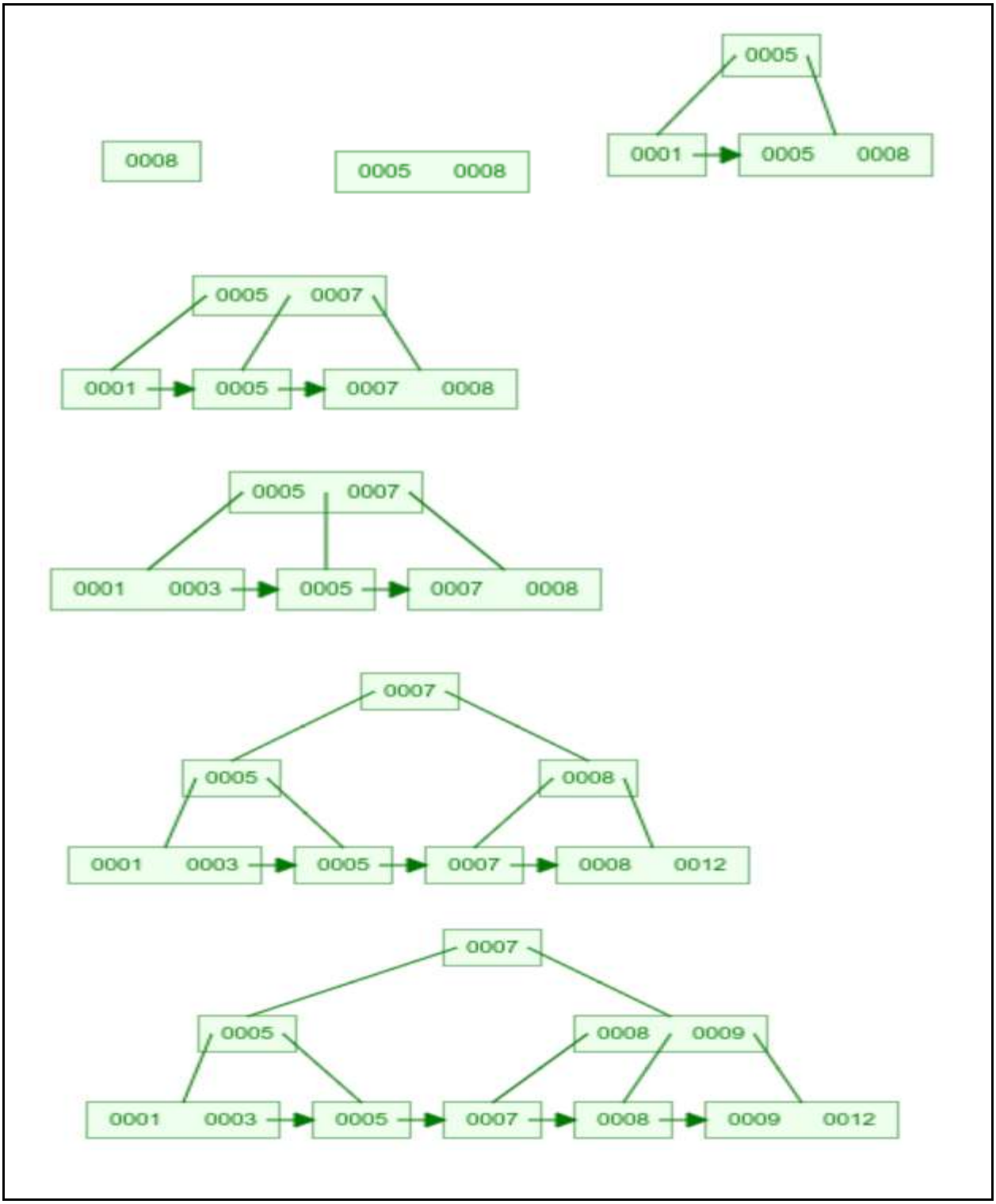
Given the B+ tree below for n= 4, delete the following values in the given sequence. After each deletion show the updated tree.

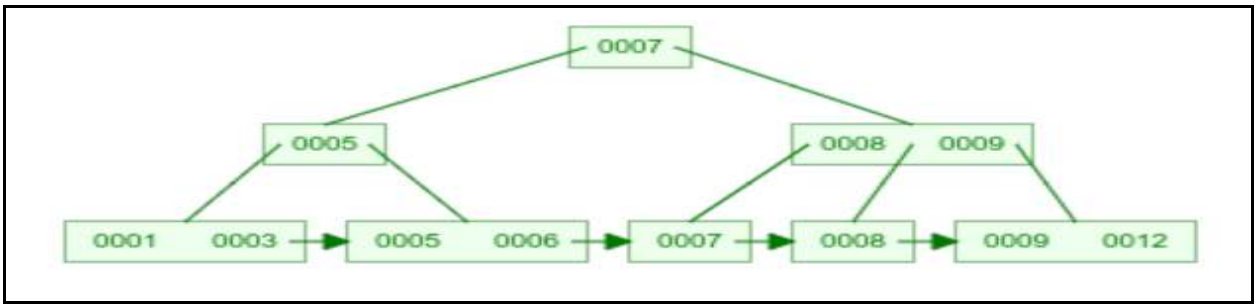
35, 90, 95, 55, 25, 30, 20, 100, 50, 50, 65, 60, 100, 115, 85



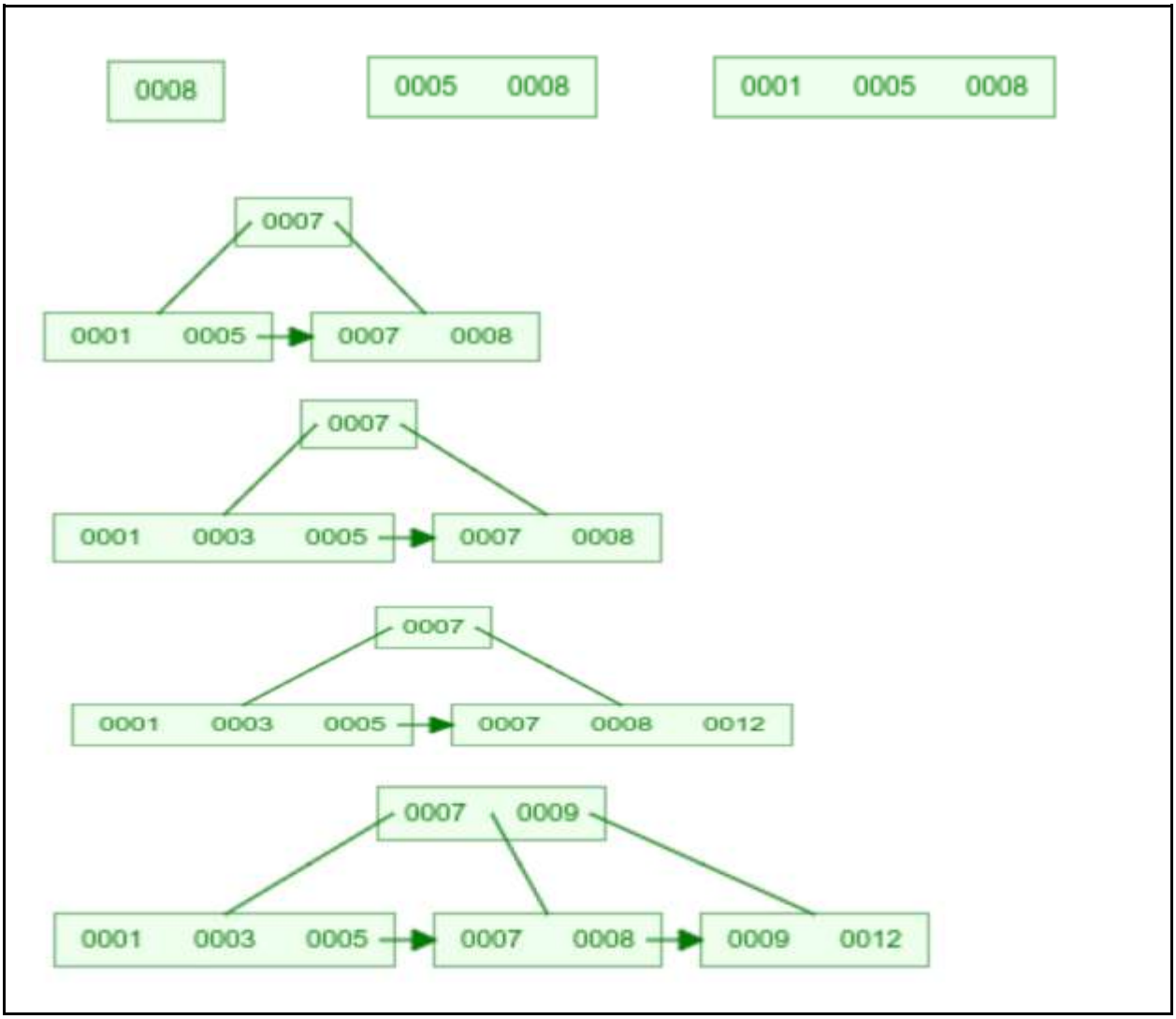
SOLUTIONS:

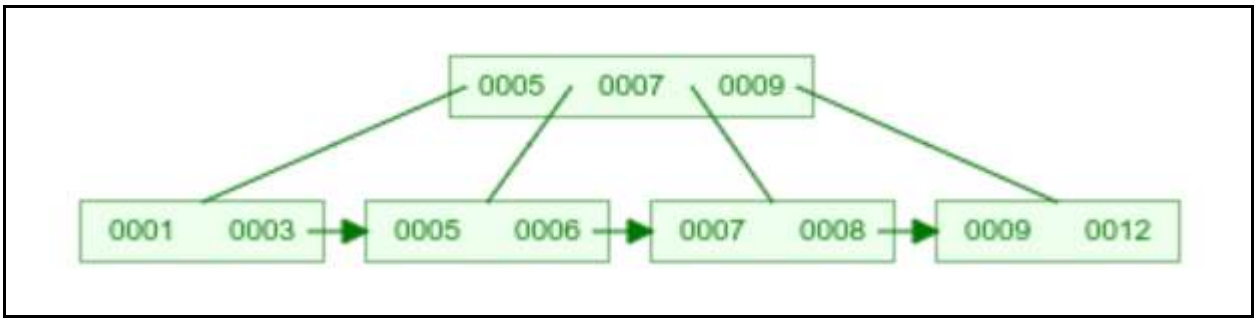
Answer 1:


0008
0005 0008
0005
0001
0005 0008
0005 0007
0001
0005
0007 0008
0005 0007
0001 0003
0005
0007 0008
0007
0005
0008
0001 0003
0005
0007
0008 0012
0007
0005
0008 0009
0001 0003
0005
0007
0008
0009 0012


0007
0005
0008
0009
0001
0003
0005
0006
0007
0008
0009
0012

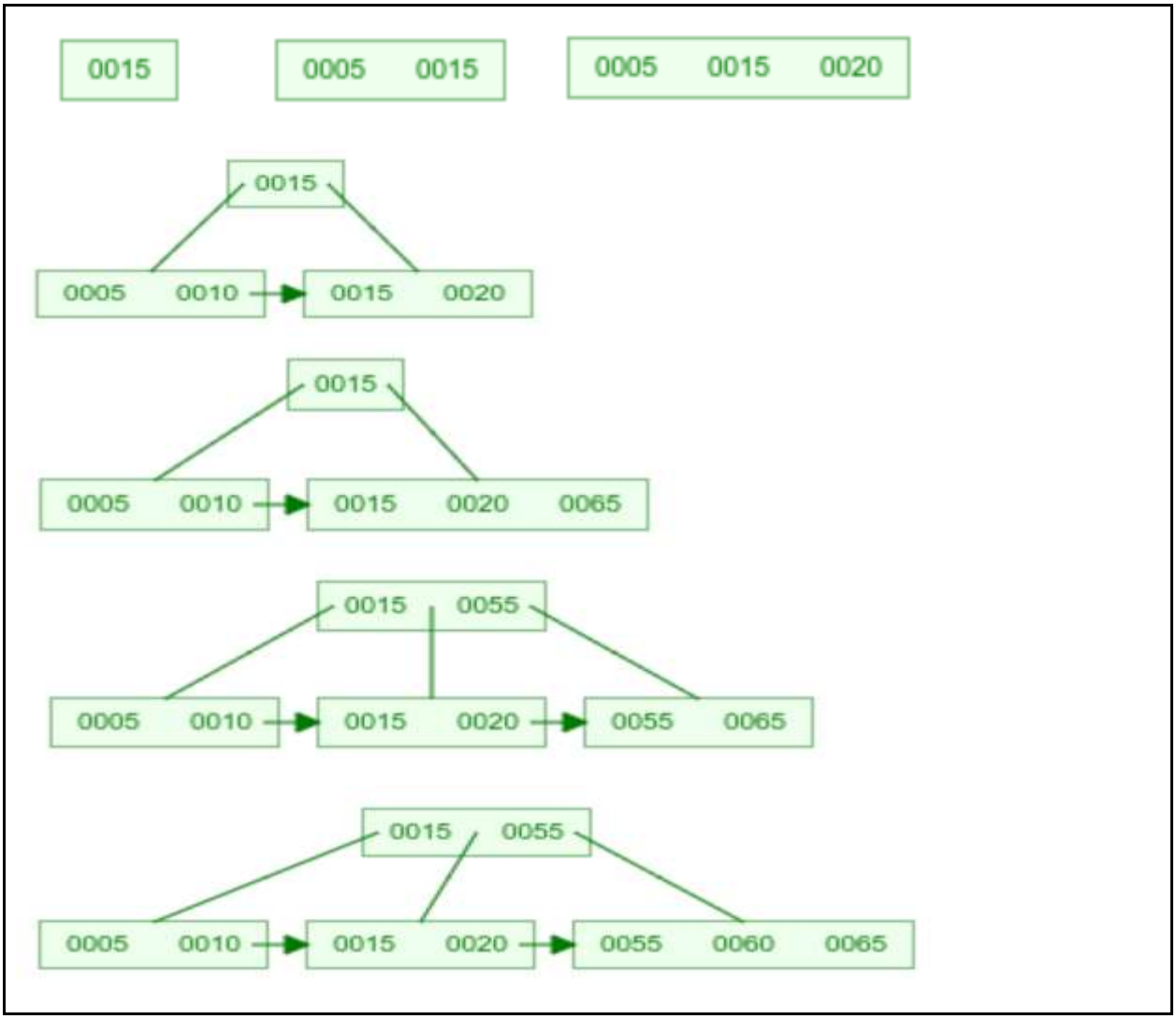
Answer 2:

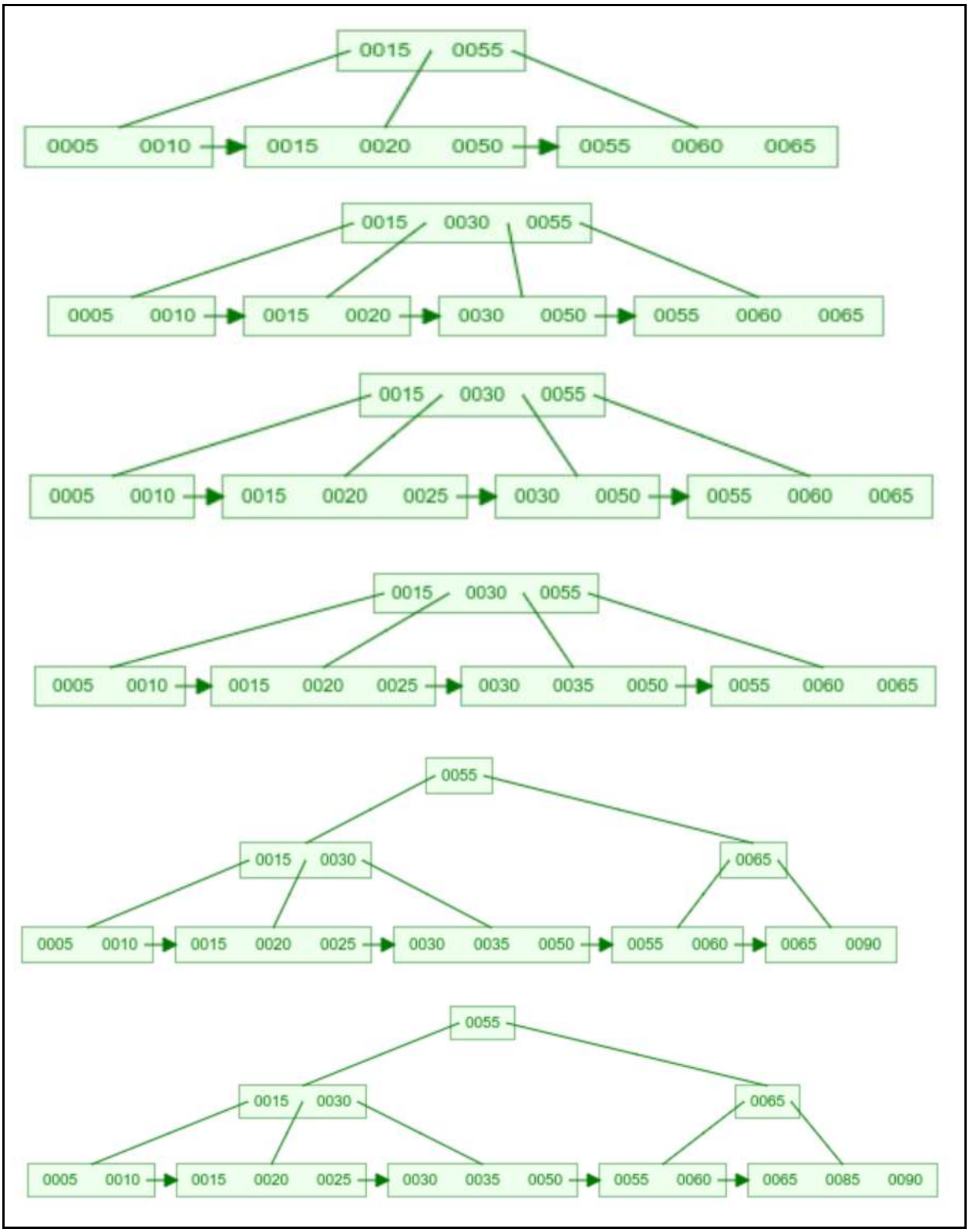

0008
0005
0008
0001
0005
0008
0007
0001
0005
0007
0008
0007
0001
0003
0005
0007
0008
0007
0001
0003
0005
0007
0008
0012
0007
0009
0001
0003
0005
0007
0008
0009
0012

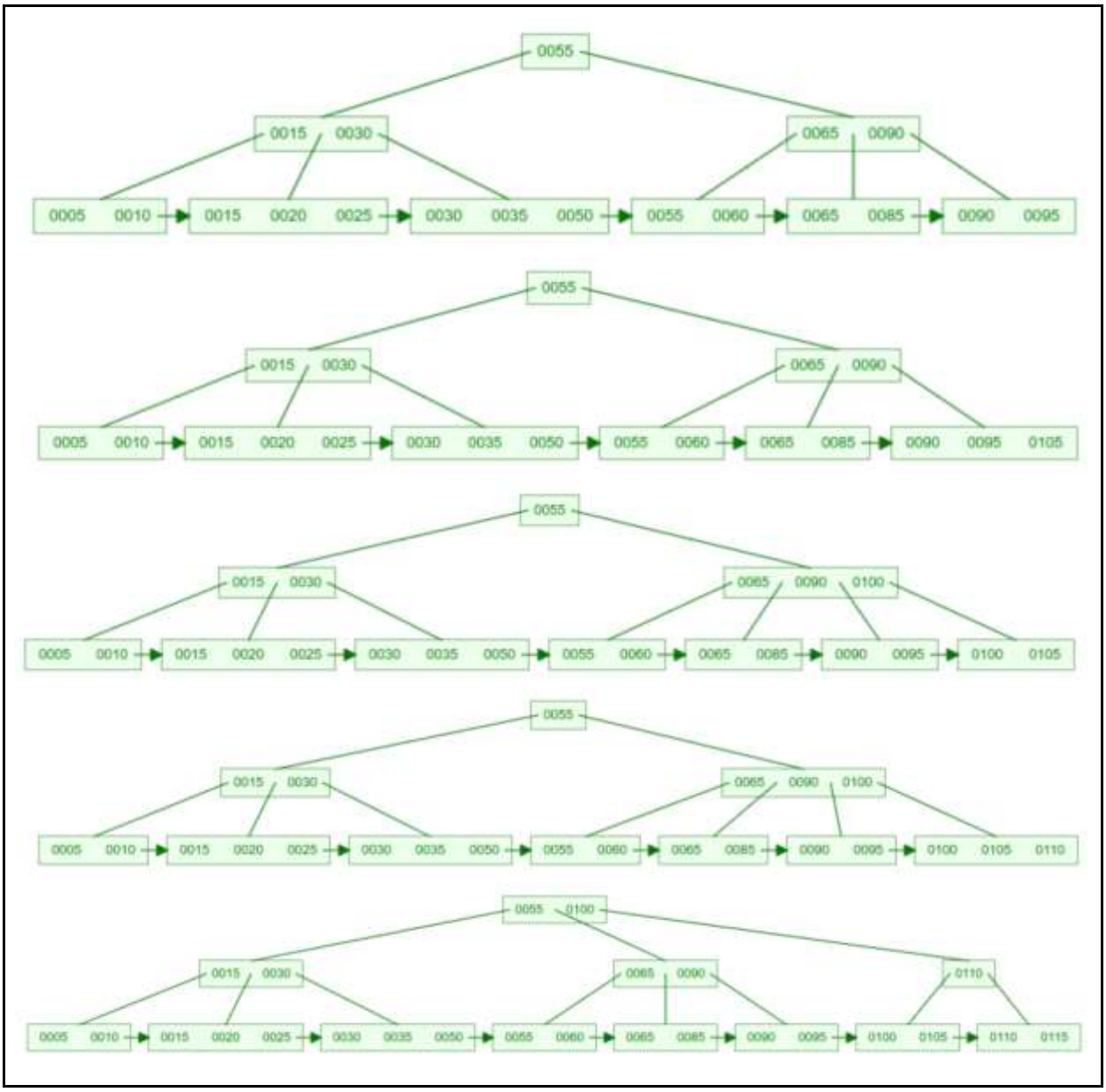

0005 0007 0009
0001 0003
0005 0006
0007 0008
0009 0012

Answer 3:

A.


0015
0005 0015
0005 0015 0020
0015
0005 0010
0015 0020
0015
0005 0010
0015 0020 0065
0015 0055
0005 0010
0015 0020
0055 0065
0015 0055
0005 0010
0015 0020
0055 0060 0065


0015 0055
0005 0010
0015 0020 0050
0055 0060 0065
0015 0030 0055
0005 0010
0015 0020
0030 0050
0055 0060 0065
0015 0030 0055
0005 0010
0015 0020 0025
0030 0050
0055 0060 0065
0015 0030 0055
0005 0010
0015 0020 0025
0030 0035 0050
0055 0060 0065
0055
0015 0030
0065
0005 0010
0015 0020 0025
0030 0035 0050
0055 0060
0065 0090
0055
0015 0030
0065
0005 0010
0015 0020 0025
0030 0035 0050
0055 0060
0065 0085 0090



B. Searching 65:

Start at root with values 55, 100. 65 is RHS of 55 which has values 65 and 90, so go to RHS of 65 and we reach the leaf, the first value is 65.

Searching 120:

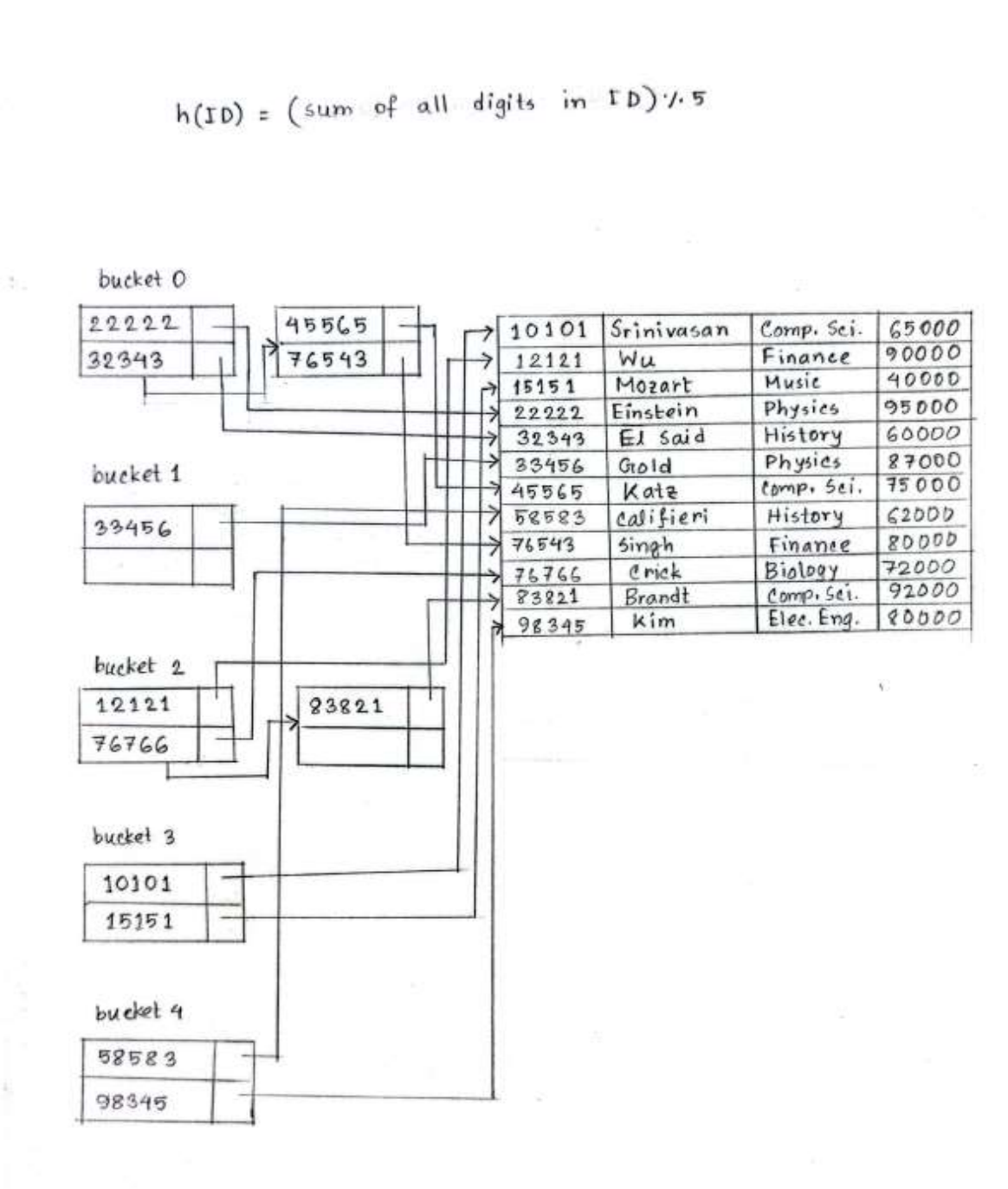
Start at root with values 55, 100. 120 is RHS of 100, where the node value is 110. 120 is again in the RHS of node 110, we reach the leaf node. The values are 110 and 115, so 120 does not exist.

Answer 4:

Take help from slides, books and online materials to write a concise answer to each question in your own words.

Answer5:

A.



B:

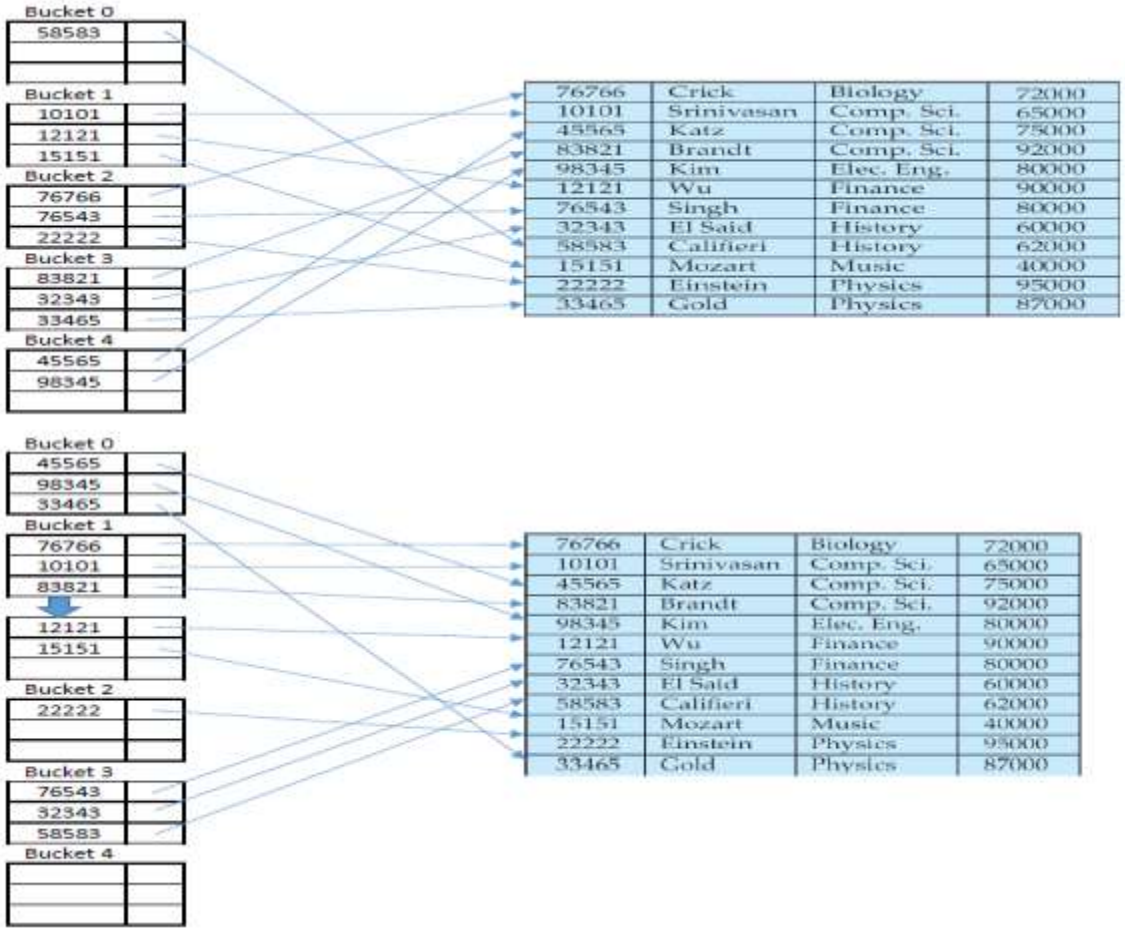
Searching for 45565:

h(45565) = 0
Therefore, we examine the buckets located at index 0. Upon retrieving this initial bucket, we encounter two entries, namely 22222 and 32343. As our intended record has not been found, we proceed to retrieve the second bucket, from which we obtain the entry 45565.

Searching for 11113:

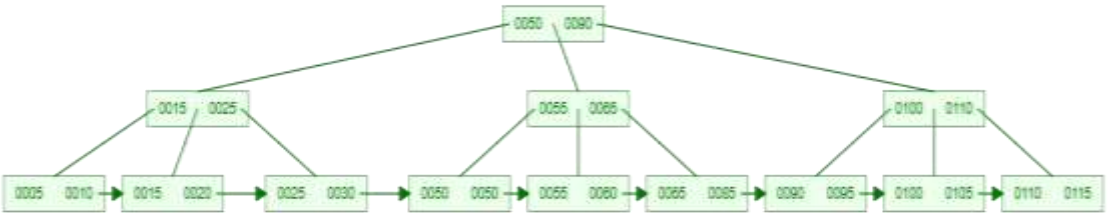
h(11113) = 2
Therefore, we examine the buckets located at index 2. Upon retrieving the initial bucket, two records, namely 12121 and 76766, are obtained. Subsequently, retrieving the second bucket yields the entry 83821. At this point, all buckets at index 2 have been retrieved, and the end of the linked list has been reached. Consequently, the search for the key 11113 concludes without success.

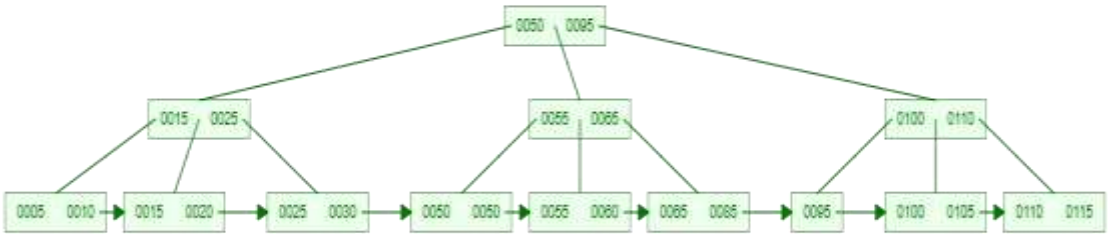
Answer 7:

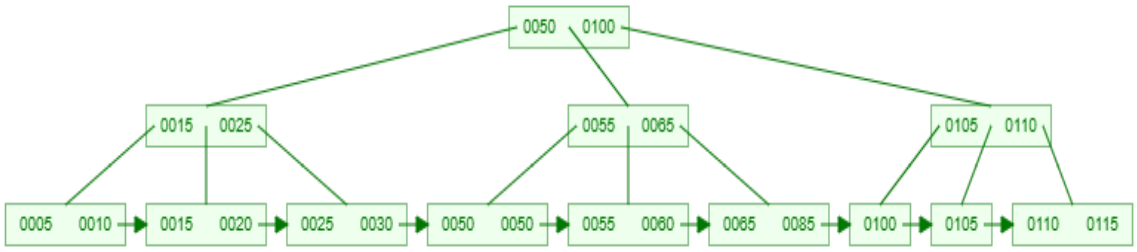


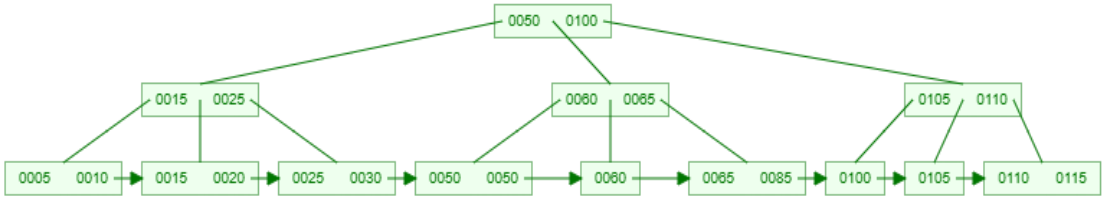
The 1st hash function is more uniform compared to the later one. As no buckets were empty in the 1st case. However, in the 2nd case not only were there empty buckets but also there was an overflow in the bucket.

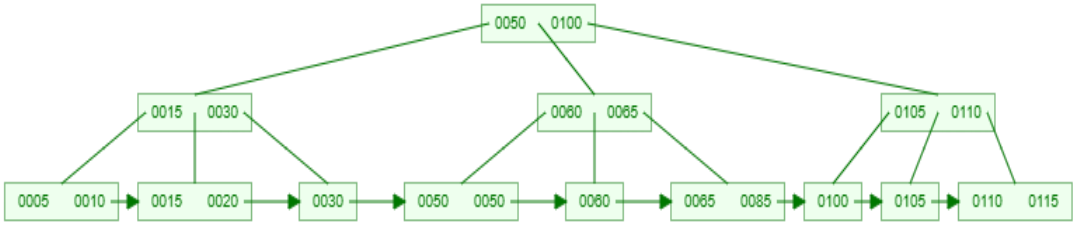
Answer 8:

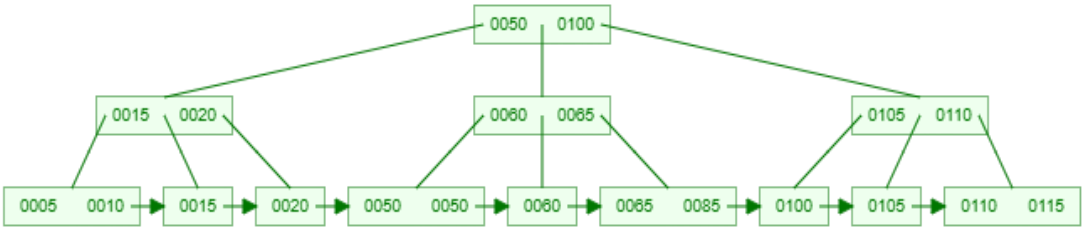

0050 0090
0015 0025
0055 0065
0100 0110
0005 0010
0015 0020
0025 0030
0050 0050
0055 0060
0065 0085
0090 0095
0100 0105
0110 0115

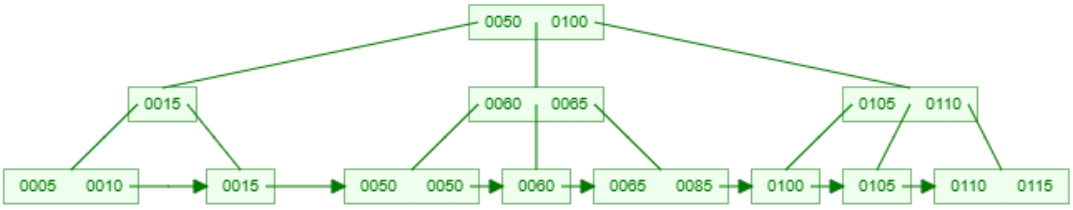

0050 0095
0015 0025
0055 0065
0100 0110
0005 0010
0015 0020
0025 0030
0050 0050
0055 0060
0065 0085
0095
0100 0105
0110 0115

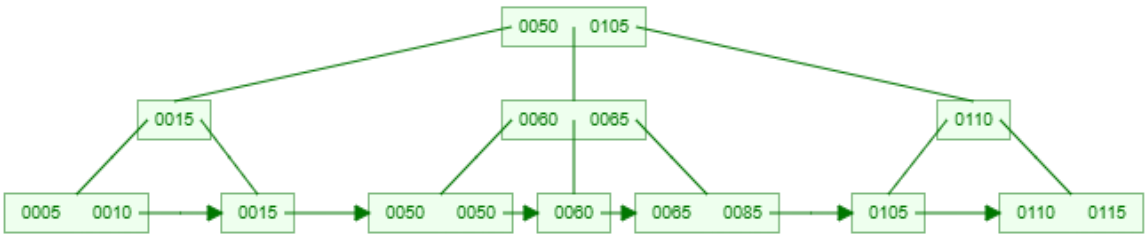

0050 0100
0015 0025
0055 0065
0105 0110
0005 0010
0015 0020
0025 0030
0050 0050
0055 0060
0065 0085
0100
0105
0110 0115

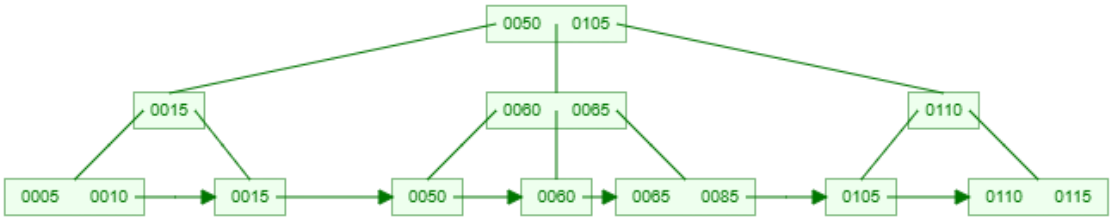

0050 0100
0015 0025
0060 0065
0105 0110
0005 0010
0015 0020
0025 0030
0050 0050
0060
0065 0085
0100
0105
0110 0115

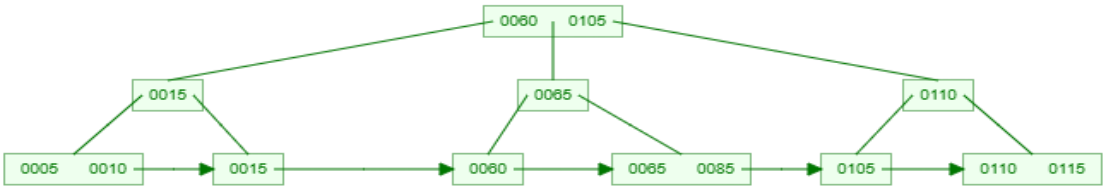

0050 0100
0015 0030
0060 0065
0105 0110
0005 0010
0015 0020
0030
0050 0050
0060
0065 0085
0100
0105
0110 0115

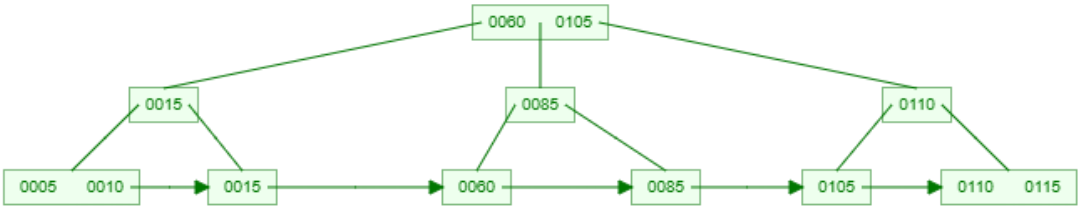

0050 0100
0015 0020
0060 0065
0105 0110
0005 0010
0015
0020
0050 0050
0060
0065 0085
0100
0105
0110 0115

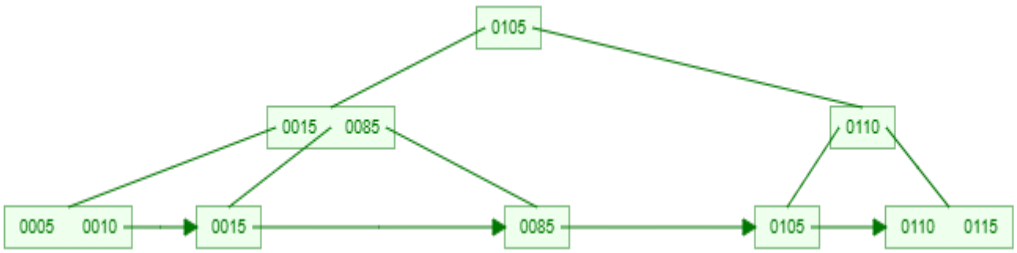

0050 0100
0015
0060 0065
0105 0110
0005 0010
0015
0050 0050
0060
0065 0085
0100
0105
0110 0115

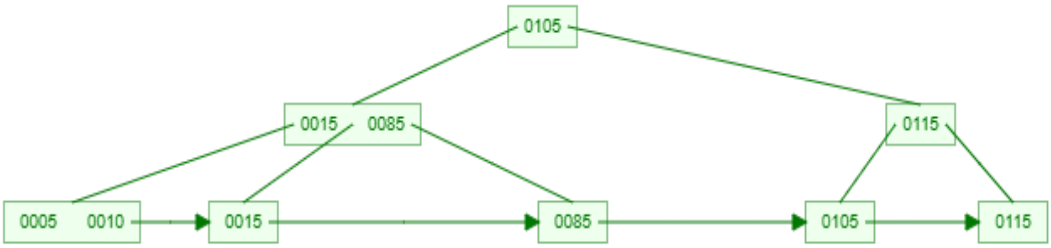

0050 0105
0015
0060 0065
0110
0005 0010
0015
0050 0050
0060
0065 0085
0105
0110 0115

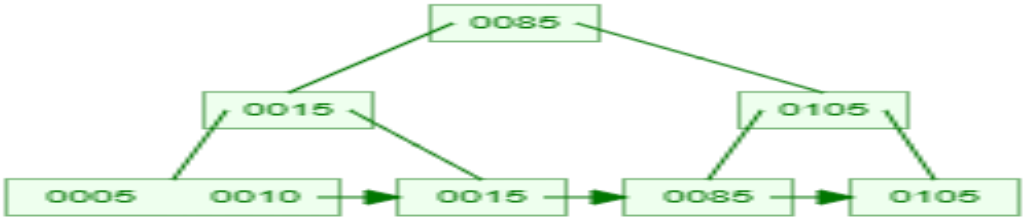

0050 0105
0015
0060 0065
0110
0005 0010
0015
0050
0060
0065 0085
0105
0110 0115

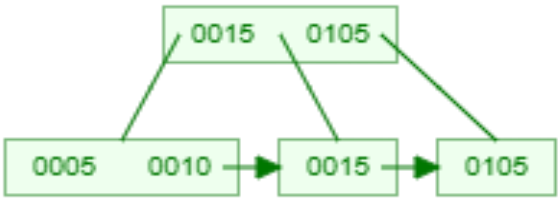

0060 0105
0015
0065
0110
0005 0010
0015
0060
0065 0085
0105
0110 0115


0060 0105
0015
0085
0110
0005 0010
0015
0060
0085
0105
0110 0115


0105
0015 0085
0110
0005 0010
0015
0085
0105
0110 0115


0105
0015 0085
0115
0005 0010
0015
0085
0105
0115


0085
0015
0105
0005 0010
0015
0085
0105


0015 0105
0005 0010
0015
0105