Exercise 3 Solution

1. What indices are suitable if a table is frequently used for finding records based on three criteria: a list of users' name, a range of users' birthday and a spatial region covering users' residence?

   Solution:

   - **Users' name**: Hash index.
   - **Users' birthday**: B+ tree index.
   - **Users' residence**: R-tree index or another spatial index such as a quadtree index.

2. Review the points on indexing with $B+$ trees. Assume a database table has 10,000,000 records and the index is built with a $B+$ tree. The maximum number of children of a node, is denoted as n. How many steps are needed to find a record if $n = 4$? How many steps are needed to find a record if $n = 100$?

   Solution:

   - When $n = 4$, the maximum height of the tree is

     $$\lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_2(10000000) \rceil = 24$$

     Therefore, 24 steps are needed.
   - When $n = 100$, the maximum height of the tree is

     $$\lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_{50}(10000000) \rceil = 5$$

     Therefore, 5 steps are needed.

3. Given the database table (shown below), build a bitmap index for the 'State' attribute.

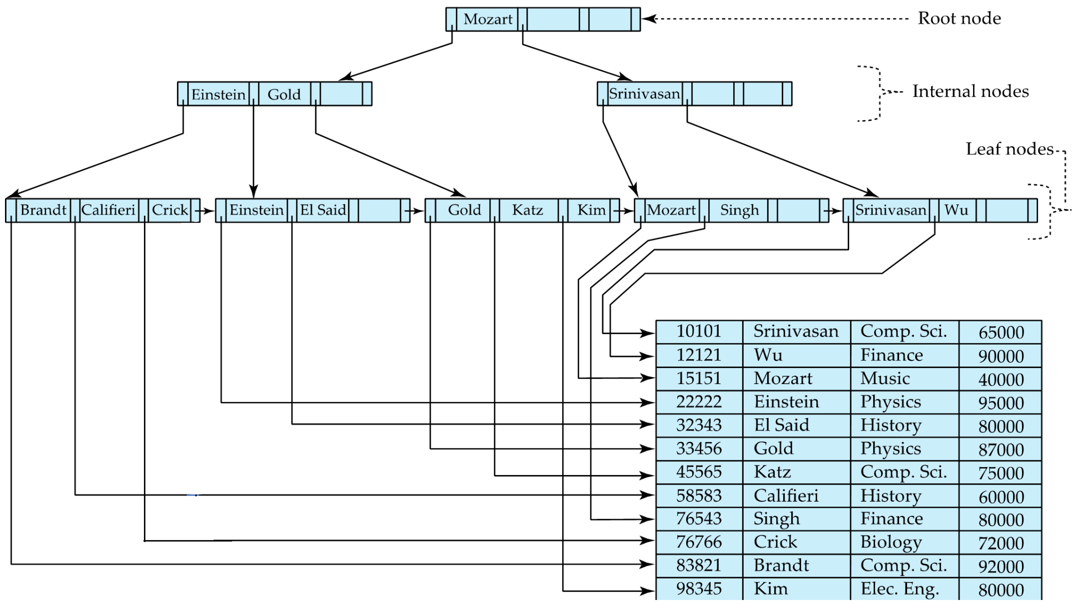
Record Num	Name	State	Income_level
0	John	VIC	L1
1	Diana	NSW	L2
2	Xiaolu	WA	L1
3	Anil	VIC	L4
4	Peter	NSW	L3

   Solution:

   Bitmap for States:

   - VIC : 10010
   - NSW : 01001
   - WA : 00100

4. Given the B+tree example below please search for "Crick" by going through the search algorithm from our lectures step by step.

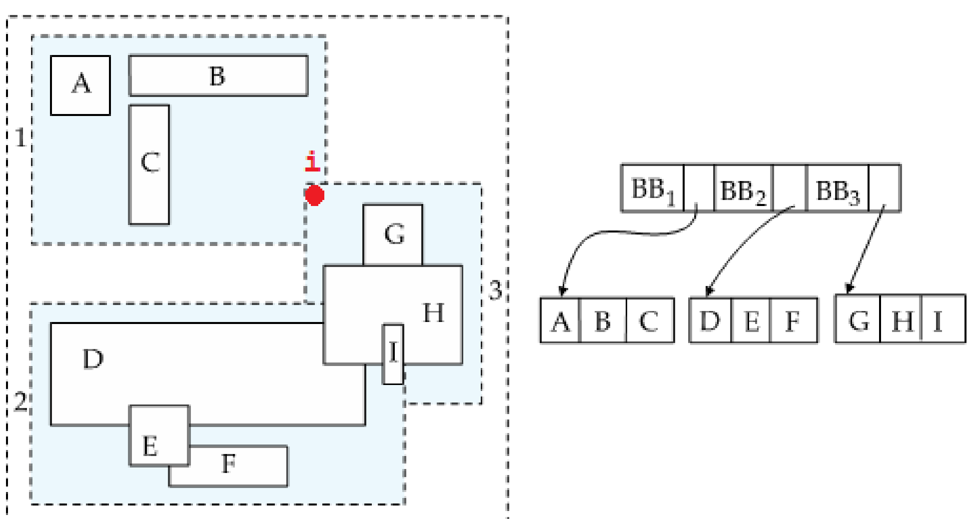


Solution:

Algorithm is on Lecture slide.

5. In the following figure, the rectangles A-I are data objects. These data objects are indexed using an R-tree, where BB1-BB3 denote bounding boxes. Given a query point location denoted as 'i' in the figure, traverse the nodes of the R-tree below in a best-first manner (that is, retrieve the node with the shortest distance from 'i' in each iteration) as discussed in class to find the 1st nearest neighbour of query point "i". If 'i' is inside a bounding box (or rectangle), the distance from i to that bounding box is zero. If there are more than one bounding boxes (or rectangles) with the same distance from i, choose the one alphabetically (e.g., i is inside both BB1 and BB3, choose BB1 first). Is there anything peculiar that you notice while traversing an R-tree?

[Note: Consider the distances just visually].



Solution:

In this traversal we first visit node BB_1 as it overlaps with the query point. The items in the priority queue is now <BB3, B, C, A>. We cannot stop at this point in the traversal as i overlaps with BB_3 as well, so we need to investigate the data there too. The items in the priority queue is now <G, H, B, C, I, A> (the exact ordering of the later ones are not important for this example, as exact distances are not given). We now find that G is the first object in this priority queue, so it is the closest object overall (that is, 1st nearest neighbour).

Due to overlaps in R-tree bounding boxes, two or more branches of an R-tree need to be traversed in many query types. In addition, as each internal node represents a bounding box, thus we are not sure about the position of objects inside a bounding box, which may necessitate that we investigate multiple bounding boxes to determine the nearest neighbour in this case.