

CS213/293 Data Structure and Algorithms 2023

IITB India

Midsem

Ashutosh Gupta

Compile date: 2023-09-16

Duration : 2 hours

Total: 60 marks

- Write all your algorithms in pseudo code
- Do not be unduly worried if you cannot finish the question paper. If you are running short of time, you may describe your algorithm idea in english. However, we deduct marks if any gaps are found.

Section A (25 marks)

1. (7 marks) Mark the following statements True / False and also provide justification.
 - (a) KMP is $O(n + m)$ for text of size n and pattern of size m .
 - (b) For a fixed array of size 2^k for integer k , the binary search always takes same amount of time in the case of unsuccessful search.
 - (c) Each black node of a red-black tree must have a red child.
 - (d) A node in a trie has at most two children.
 - (e) Evaluating an arithmetic expression stored in a tree needs preorder walk.
 - (f) A hash function must be one-to-one.
 - (g) A trie for n words has n^2 leaves.
2. (4 marks) Give a tree, if exists, that is binary search tree, is heap, and has more than two nodes. If such a tree does not exist, give reason.
3. (7 marks) Give pseudo code of implementation of queue using circular linked list. You need to give implementation of enqueue and dequeue.
4. (7 marks) Given two heaps give an efficient algorithm to merge the heaps.

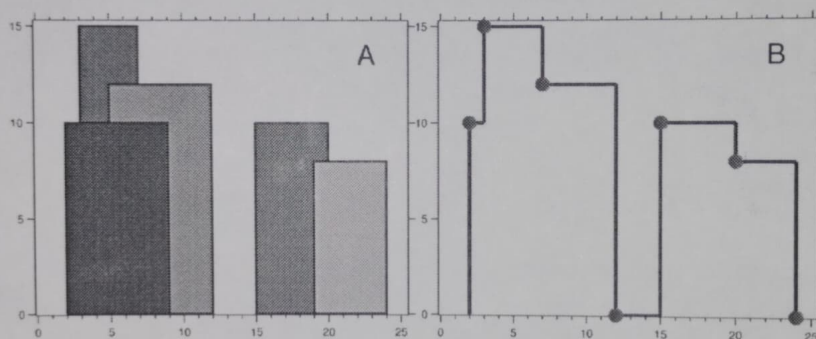
Section B (35 marks)

5. (10 marks) Given a binary tree labeled with integers, we define maximum sum path as a path from a leaf node to other leaf node with the maximum sum. Give pseudocode to print maximum sum path. If multiple such path exists print any one of those. Ensure that time complexity is $O(n)$, where n is number of nodes in the tree.

6. (10 marks) A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Give a space and time efficient algorithm for computing skyline formed by the buildings collectively given the locations and heights of all the buildings.

The geometric information of each building is given in the array `buildings` where `buildings[i] = [lefti, righti, heighti]`: `lefti` is the x coordinate of the left edge of the *i*th building, `righti` is the x coordinate of the right edge of the *i*th building, `heighti` is the height of the *i*th building.

You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0. The skyline should be represented as a list of "key points" sorted by their x-coordinate in the form `[[x1,y1],[x2,y2],...]`. Each key point is the left endpoint of some horizontal segment in the skyline except the last point in the list, which always has a y-coordinate 0 and is used to mark the skyline's termination where the rightmost building ends. Any ground between the leftmost and rightmost buildings should be part of the skyline's contour. Note: There must be no consecutive horizontal lines of equal height in the output skyline. For instance, `[...,[2 3],[4 5],[7 5],[11 5],[12 7],...]` is not acceptable; the three lines of height 5 should be merged into one in the final output as such: `[...,[2 3],[4 5],[12 7],...]` Example: Input: `buildings = [[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]` Output: `[[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]]`.



(Image is borrowed from internet.)

7. (15 marks) Design a data structure such that there are k sets of integers and the total number of integers across all k sets is N . An integer may be present in multiple sets. Each set contains at least one integer. Given an query integer e , we have to find the closest integer to query e in each of k sets. Closeness between two integers i, j is defined as: $|i - j|$. Give a space and time efficient algorithm for the query operation. You may pre-process of the input sets.

Example: For sets $\{\{6, 8\}, \{8, 2, 5\}\}$, and $e = 4$, we return $(6, 2)$.