

RadixSort. Arbori binari de căutare

Bibliografie

- <https://web.stanford.edu/class/archive/cs/cs161/cs161.1168/lecture8.pdf> (similar Cormen)
- <https://www.geeksforgeeks.org/binary-search-tree-data-structure/?ref=lbp>
- <https://algs4.cs.princeton.edu/32bst/> - si cartea Algorithms, <https://algs4.cs.princeton.edu/lectures/keynote/32BinarySearchTrees.pdf>

Probleme

1. **Radix Sort** <https://www.pbinfo.ro/probleme/2700/radixsort>
2. Implementați operațiile de bază pentru un arbore binar de căutare **recursiv/nerecursiv**
 - Inserarea unei valori date
<https://leetcode.com/problems/insert-into-a-binary-search-tree/>
<https://www.techiedelight.com/?problem=InsertKeyIntoBST>
 - Parcurgerea în inordine
<https://www.pbinfo.ro/probleme/3010/bst>
 - Căutarea unei valori date
<https://www.techiedelight.com/?problem=SearchKeyInBST>
 - Determinarea valorii minime și maxime din arbore
 - Determinarea celei mai apropiate valori mai mare/mai mică decât o valoare dată (succesor / predecesor în inordine)
<https://www.techiedelight.com/?problem=InorderPredecessorBST>
<https://www.techiedelight.com/?problem=InorderSuccessorBST>
 - Ștergerea valorii minime
 - Ștergerea unei valori date
<https://leetcode.com/problems/delete-node-in-a-bst/>
<https://www.techiedelight.com/?problem=DeleteKeyFromBST>
3. <https://leetcode.com/problems/binary-tree-preorder-traversal/> și nerecursiv – folosind stiva
4. <https://leetcode.com/problems/range-sum-of-bst/description/>
5. <https://leetcode.com/problems/maximum-depth-of-binary-tree/>
6. <https://leetcode.com/problems/balanced-binary-tree/>
7. leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree/
8. <https://www.techiedelight.com/?problem=LowestCommonAncestorII> (similar cu 6, dar aici nodurile pot sa nu fie în arbore)