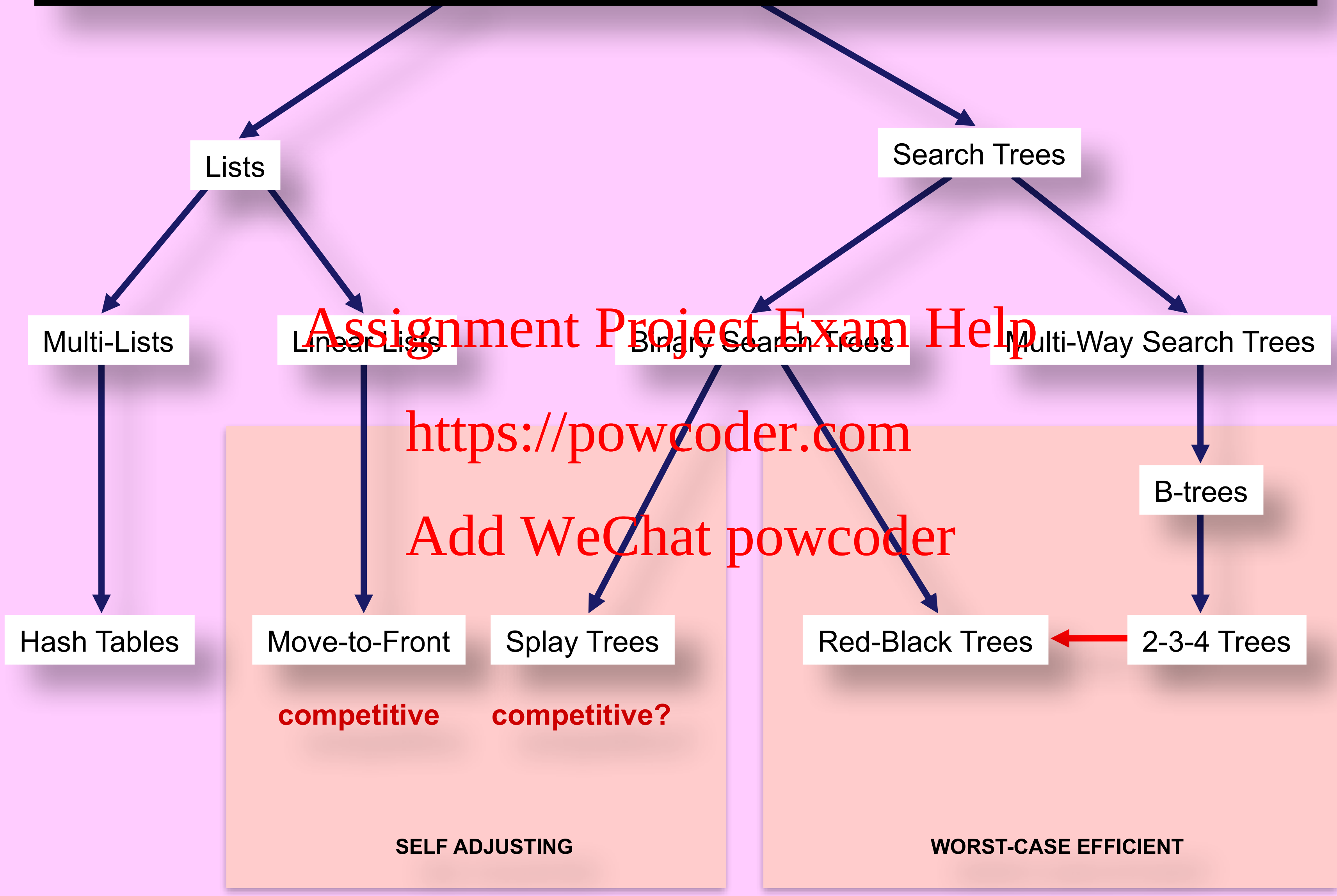
EECS 4101/5101

Prof. Andy Mirzaian

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Search Trees

DICTIONARIES



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

TOPICS

- Binary Trees
- Binary Search Trees
- Multi-way Search Trees

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

References:

- [CLRS] chapter 12

Assignment Project Exam Help

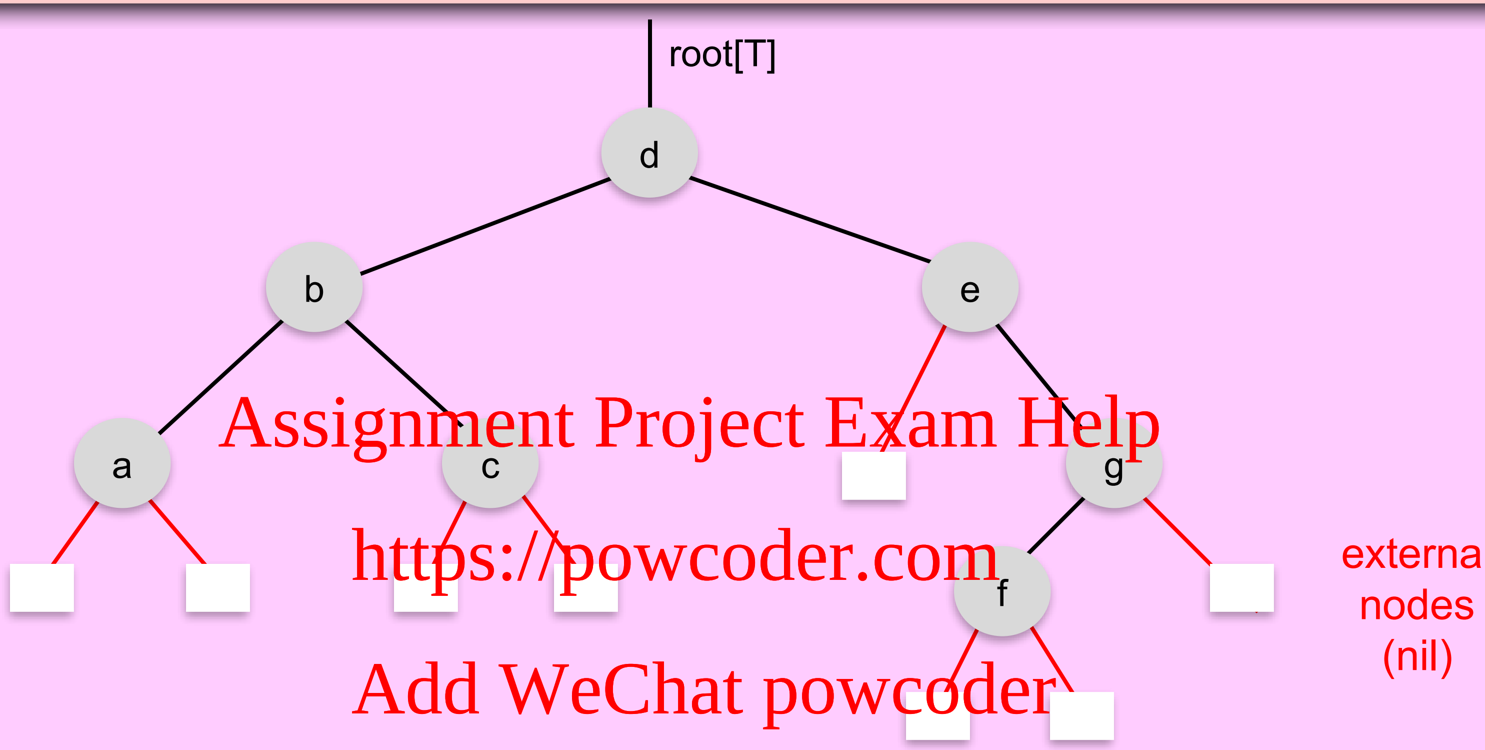
https://powcoder.com

Add WeChat powcoder

Binary Trees

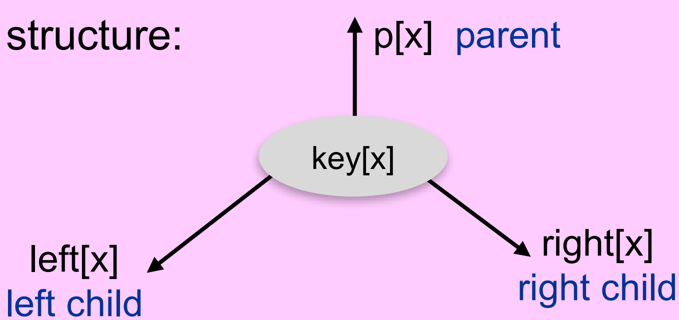
Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Binary Trees: A Quick Review



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

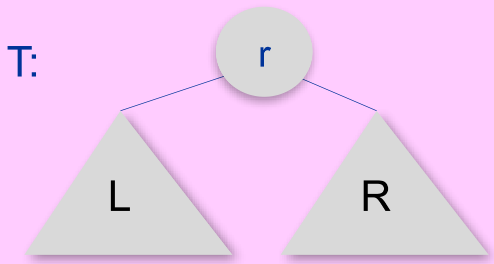
Node x structure:

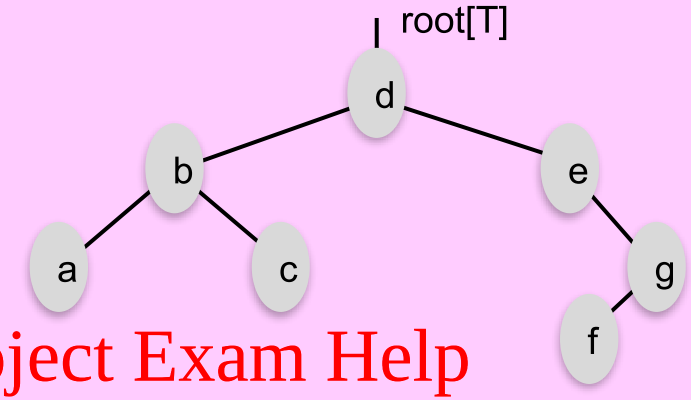


n (internal) nodes
n-1 (internal) edges

n+1 external nodes (nil)
n+1 external edges (nil)

Binary Tree Traversals





Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Inorder(T): Inorder(L); r; Inorder(R). abcdefg
  - Preorder(T): r ; Preorder(L); Preorder(R). dbacegf
- Postorder(T): Postorder(L); Postorder(R); r. acbfged

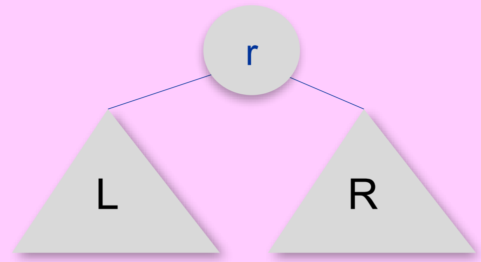
graph DFS

  - Levelorder(T): non-decreasing depth order dbeacgf → graph BFS

    (same depth left-to-right)

Traversals in O(n) time

```
procedure Inorder(x)
1.        if  x = nil  then return
2.        Inorder(left[x])
3.        visit(x)
4.        Inorder(right[x])
end
```



Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Running Time Analysis by Accounting:

Line 1: n+1 external nodes (return), n (internal) nodes (continue).

Line 3: Assume visit takes O(1) time.

Lines 2 & 4: After recursive expansion:

Each node x (internal or external) visited exactly once.

O(1) time execution of lines 1 & 3 charged to node x.

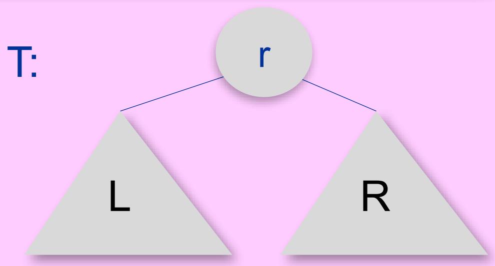
Total n + (n+1) nodes, each charged O(1) time.

Total time = O(2n+1) = O(n).

- Preorder and Postorder are similar and take O(n) time.
- Exercise: Write a simple O(n) time algorithm for Levelorder. [Hint: use a queue.]

Running Time Analysis by Recurrence

$$Time(T) = \begin{cases} Time(L) + Time(R) + 1 & \text{if } T \neq \text{nil} \\ 1 & \text{if } T = \text{nil} \end{cases}$$



CLAIM: Time(T) = 2 |T| + 1.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Proof: By induction on |T|.

Basis (|T|=0): Time(T) = 1 = 2 |T| + 1.

Induction Step (|T| > 0):

Time(T) = Time(L) + Time(R) + 1 [by the recurrence]
= (2|L|+1) + (2|R|+1) + 1 [by the Induction Hypothesis]
= 2(|L| + |R| + 1) + 1
= 2 |T| + 1.

Binary Search Trees

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

BST from Binary Search on Sorted Array

E0 < K1 < E1 < K2 < E2 < K3 < E3 < K4 < E4 < K5 < E5 < K6 < E6 < ••• < Kn < En

E0 K1 E1 K2 E2 K3 E3 K4 E4 K5 E5 K6 E6

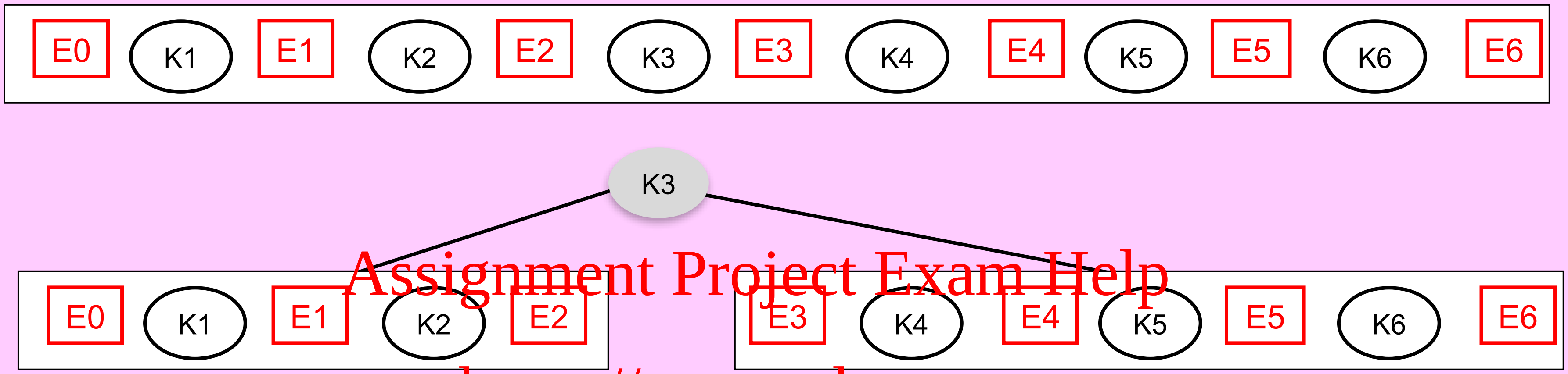
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

BST from Binary Search on Sorted Array

E0 < K1 < E1 < K2 < E2 < K3 < E3 < K4 < E4 < K5 < E5 < K6 < E6 < ••• < Kn < En



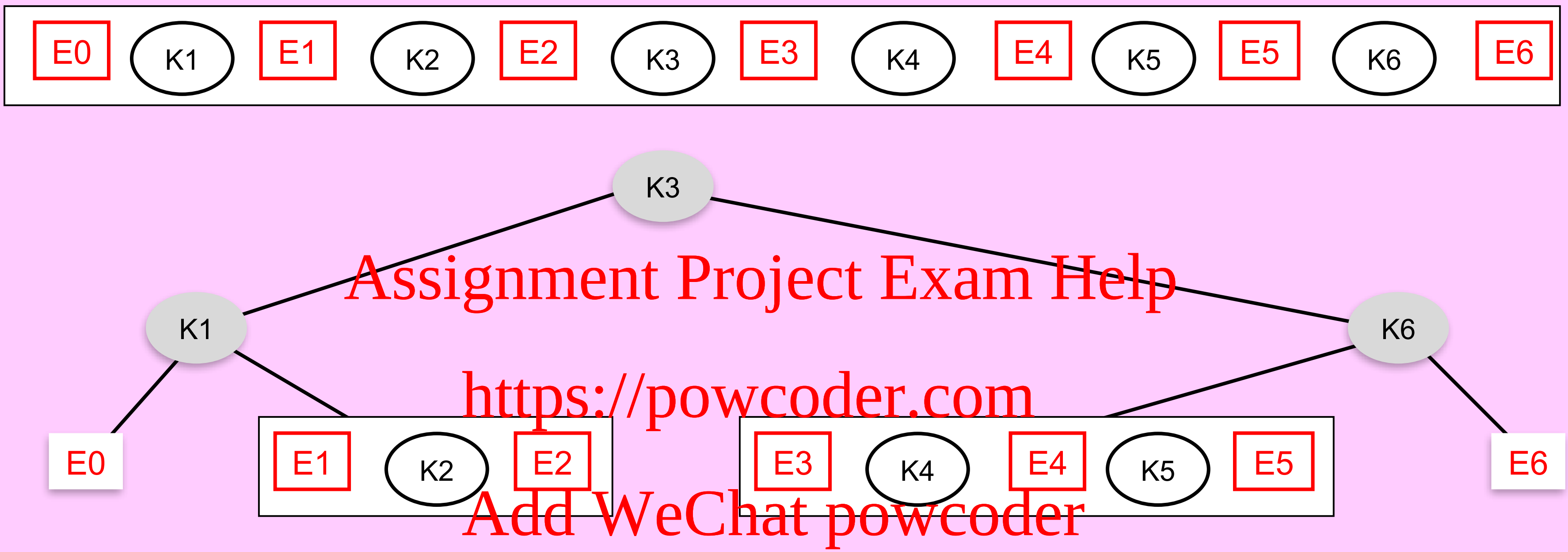
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

BST from Binary Search on Sorted Array

E0 < K1 < E1 < K2 < E2 < K3 < E3 < K4 < E4 < K5 < E5 < K6 < E6 < ••• < Kn < En



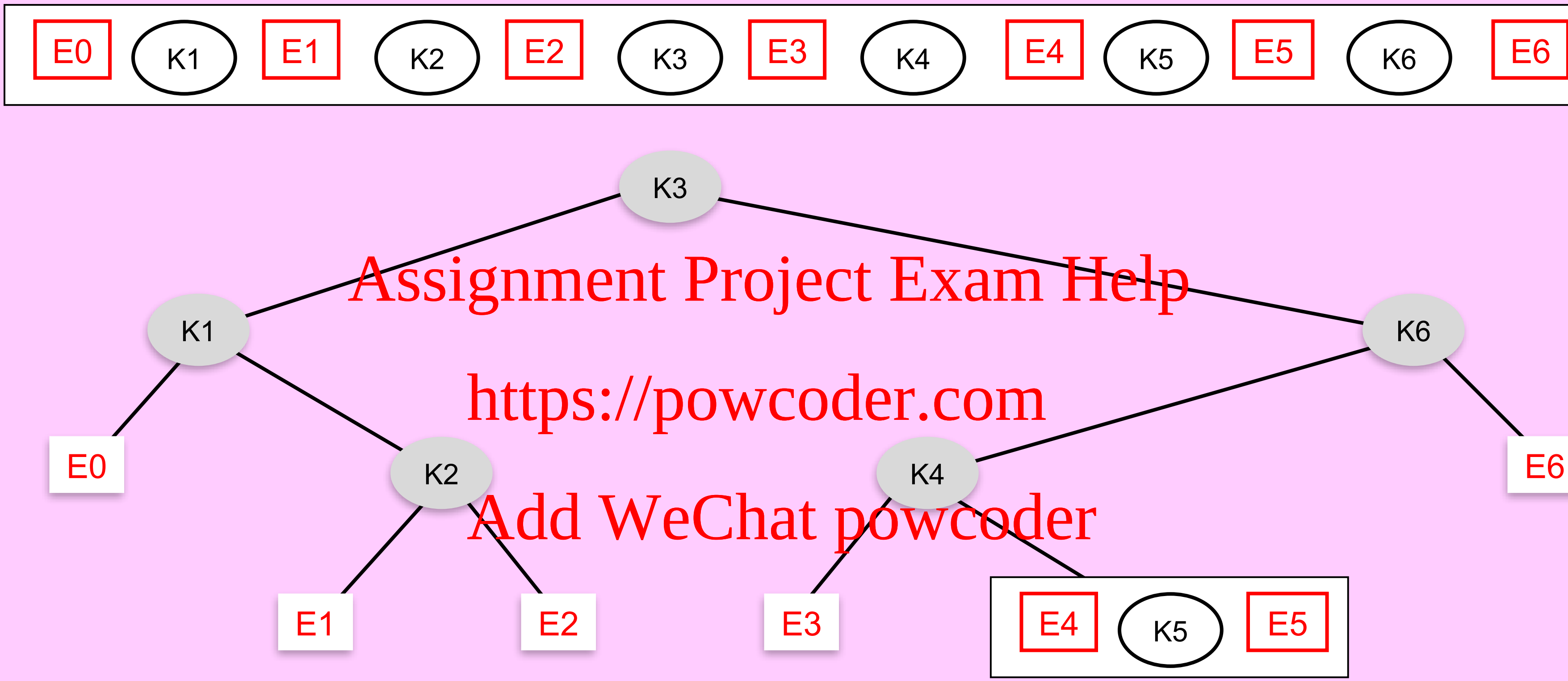
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

BST from Binary Search on Sorted Array

E0 < K1 < E1 < K2 < E2 < K3 < E3 < K4 < E4 < K5 < E5 < K6 < E6 < ••• < Kn < En



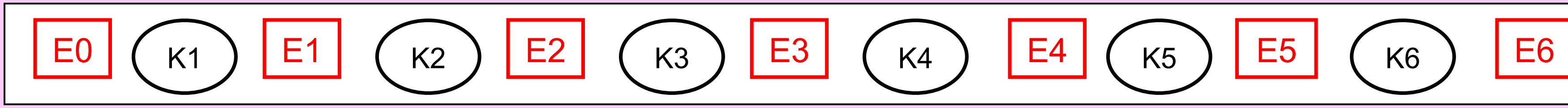
Assignment Project Exam Help

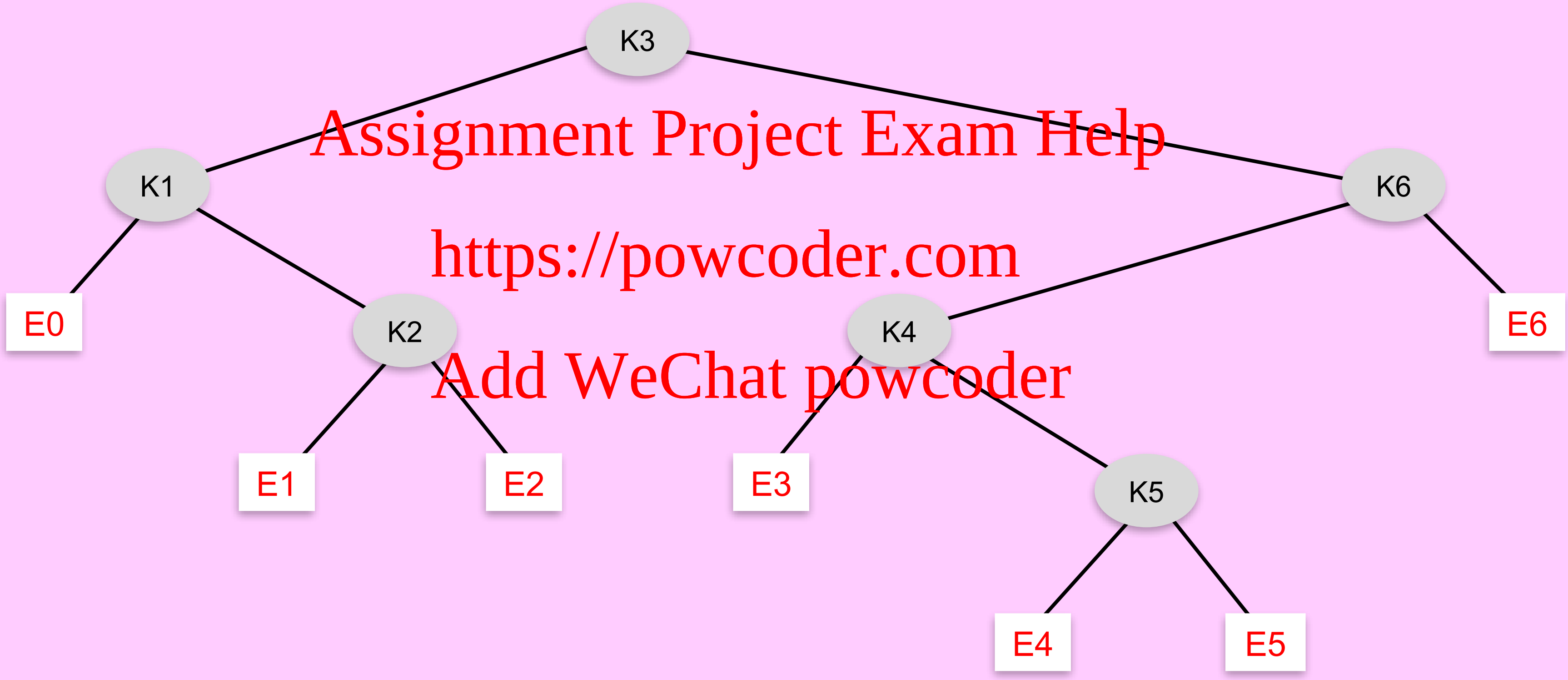
https://powcoder.com

Add WeChat powcoder

BST from Binary Search on Sorted Array

E0 < K1 < E1 < K2 < E2 < K3 < E3 < K4 < E4 < K5 < E5 < K6 < E6 < ••• < Kn < En





Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

SORTED ORDER ≡ BST INORDER

BST Definition

BST is a binary tree T with one distinct key per node such that:

- Inorder node sequence of T encounters keys in sorted order.

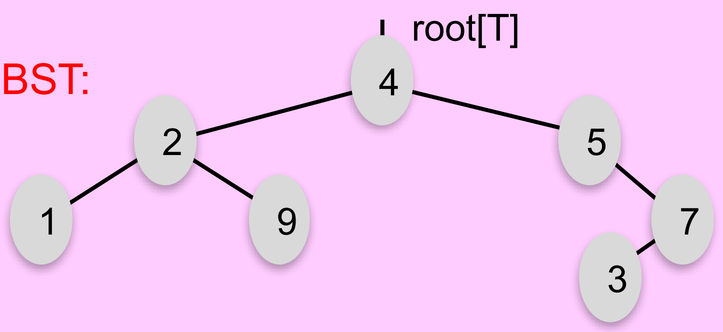
- Equivalent definition: For all nodes x & y in T:
  - If x is in the left subtree of y, then key[x] < key[y], and
  - If x is in the right subtree of y, then key[x] > key[y].

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- Wrong definition: For all nodes x & y in T:
  - If x is left child of y, then key[x] < key[y], and
  - If x is right child of y, then key[x] > key[y].

  necessary but not sufficient

Not a BST:



Path following routines

- Search(K,x): access the (possibly external) node with key K in the BST rooted at x.
- Insert(K,x): insert key K in the BST rooted at x. (No duplicates.)
- Delete(K,x): delete key K from the BST rooted at x.

Some auxiliary routines:

- Minimum(x): find the minimum key node in the BST rooted at x.
- Maximum(x): find the maximum key node in the BST rooted at x.
- Predecessor(x,T): find the Inorder predecessor of node x in binary tree T.
- Successor(x,T): find the Inorder successor of node x in binary tree T.

use parent pointers

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

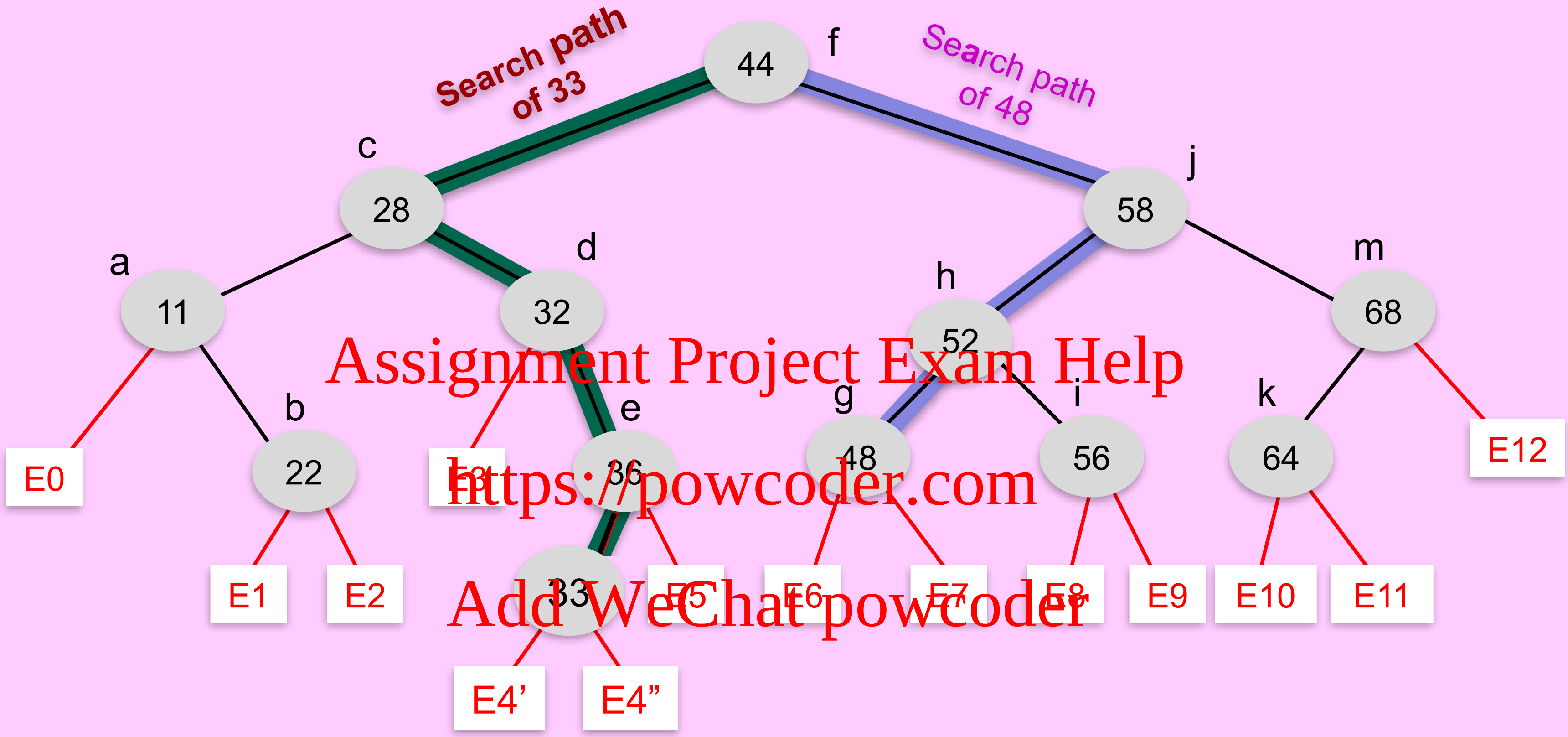
- Dictionary: Search, Insert, Delete
- Priority Queue: DeleteMin (or DeleteMax), Insert

These operations take O(h) time.

h = height of the tree.

$\lfloor \log n \rfloor \le h < n$.

Examples

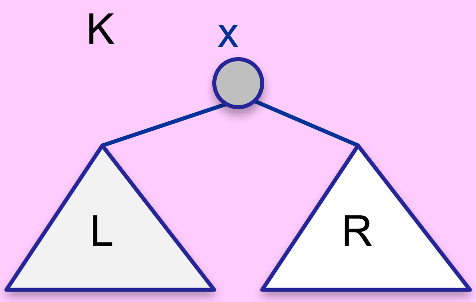


Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Search (48)	Predecessor (c)	Minimum (i)
Search (33)	Successor (b)	Maximum (c)
Insert (33)	Predecessor (a)	Minimum (a)
Delete (32)	Predecessor (f)	Minimum (f)
Delete (58)	Successor (e)	Maximum (f)

Search

```
procedure Search(K,x)
1.     If  x = nil      then return nil
2.     if K = key[x] then return x
3.     if K < key[x] then return Search(K, left[x])
4.     if K > key[x] then return Search(K, right[x])
end
```

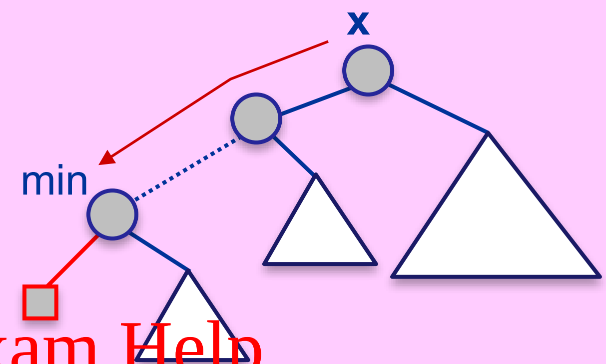


Assignment Project Exam Help

Running Time:
https://powcoder.com
We spend O(1) time per node, going down along the search path of K.
Total time = O(length of search path of K) = O(h).
Add WeChat powcoder

Minimum & Maximum

```
procedure Minimum(x)
    if  x = nil  then return nil
    y ← x
    while  left[y] ≠ nil  do y ← left[y]
    return y
end
```



Assignment Project Exam Help

https://powcoder.com

Maximum is left-right symmetric. Follow rightmost path down from x.

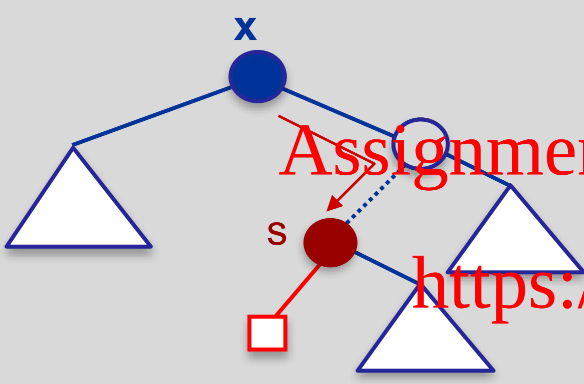
Add WeChat powcoder

Running Time of Minimum (resp., Maximum):
We spend O(1) time per node along the leftmost (resp., rightmost) path down from x.
Total time = O(h).

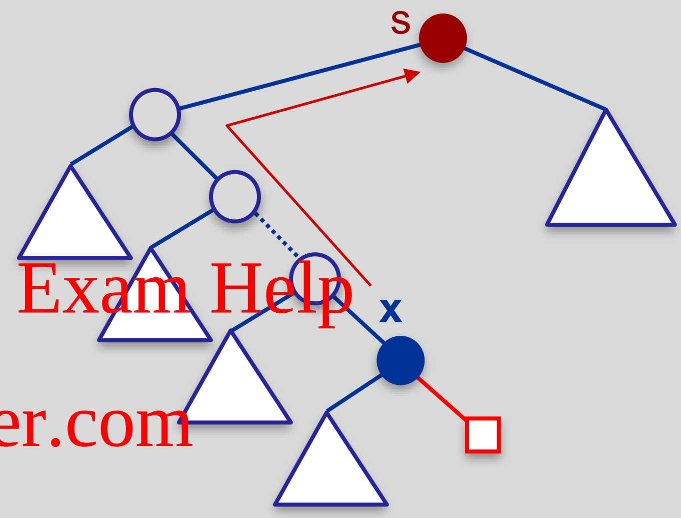
Successor & Predecessor

Find s = successor of x.

case 1: right[x] ≠ nil.
s is min of right subtree of x.



case 2: right[x] = nil.
x is max of left subtree of s.



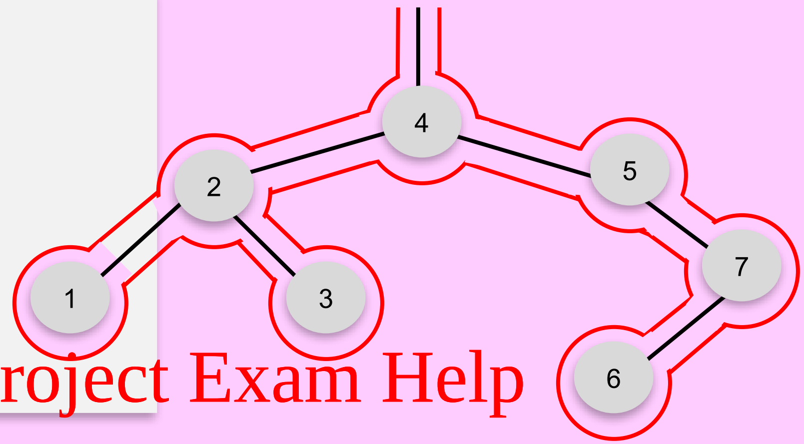
Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

```
procedure Successor(x, T)
      if  right[x] ≠ nil then return Minimum(right[x])
      y ← x
      while  p[y] ≠ nil  and y = right[p[y]] do  y ← p[y]
      return p[y]
end
```

Predecessor is symmetric. **Running Time:** O(h).

Non-recursive Inorder

```
ocedure Inorder(T)
  x ← Minimum(root[T])
  while  x ≠ nil  do
               visit (x)
               x ← Successor(x, T)
  end-while
d
```



Assignment Project Exam Help

Running Time: Minimum & Successor are called O(n) times, each time taking O(h) time. Is the total O(nh) time?

https://powcoder.com

It's actually O(n) time total: each of O(n) edges of the tree are traversed twice (once down, once up). Why?

Add WeChat powcoder

Also can do amortized analysis using stack with multipop analogy.

See Exercise 8:
- This linear-time non-recursive Inorder procedure uses parent pointers.
- If parent pointers are not available, one could maintain a stack of the ancestral nodes of x. Fill in the details.
- Write a linear-time non-recursive in-place Inorder procedure without parent pointers. (In-place means you cannot use any stack or equivalent; use just the given tree and O(1) additional scalar variables.)

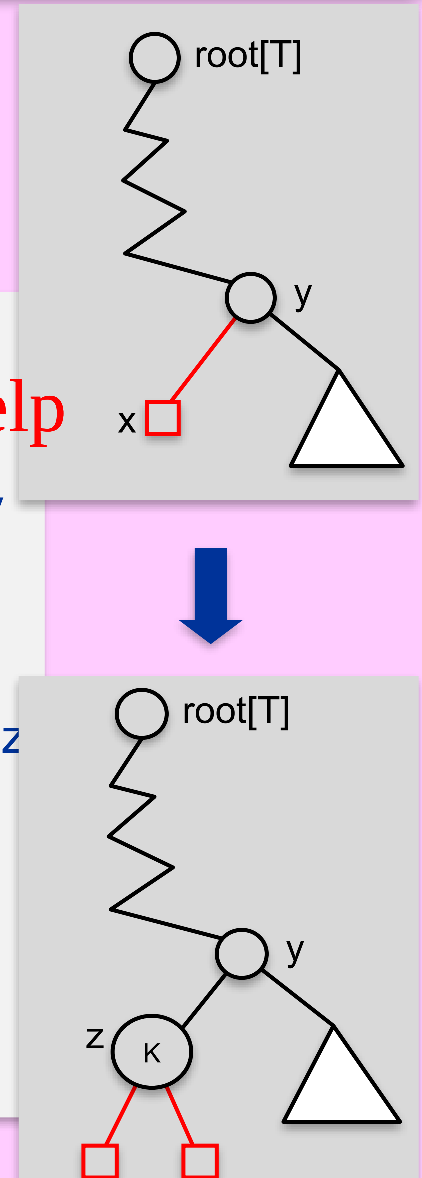
Insert

```
procedure Insert(K,T)
1.        AuxInsert(K, T, root[T], nil)
end
```

```
procedure AuxInsert(K,T,x,y)        (* y = parent of x *)
1a.       if  x = nil  then do
1b.                z  ← a new node
1c.                key[z] ← K; left[z] ← right[z] ← nil;  p[z] ← y
1d.                if y = nil  then root[T] ← z
1e.                             else if  K < key[y]
1f.                                    then left[y] ← z
1g.                                    else right[y] ← z
1h.                return
1i.       end-if

2.        if K < key[x]  then  AuxInsert(K, T, left[x], x)
3.        if K > key[x]  then  AuxInsert(K, T, right[x], x)
end
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



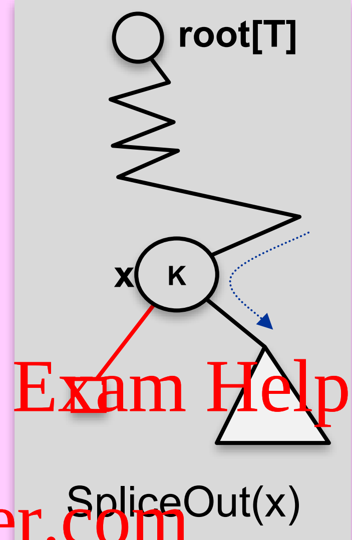
Running Time: O(length of search path of K) = O(h).

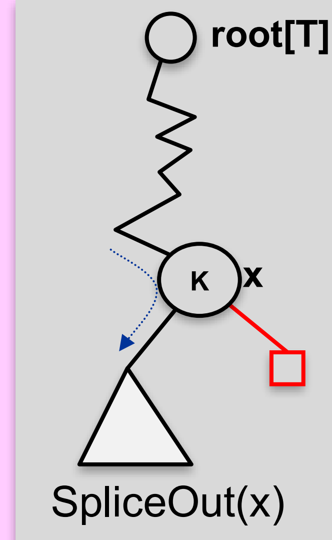
Delete

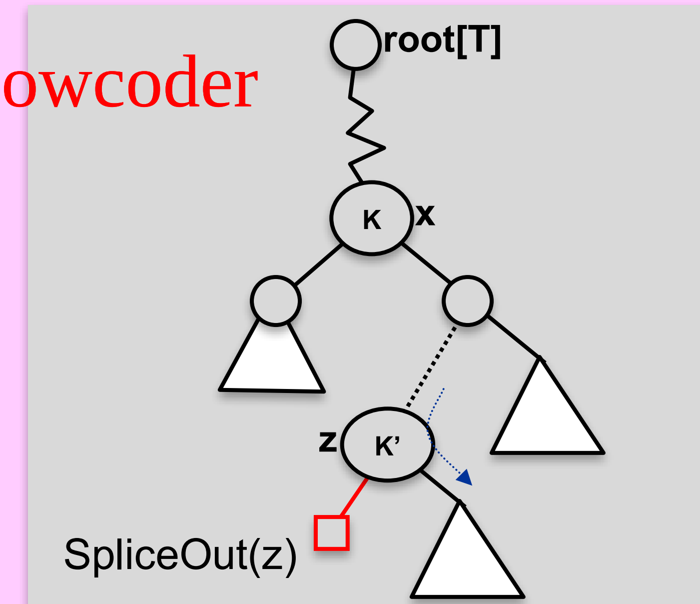
procedure Delete(K,T)
1. x ← Search(K,T)
2. **if** x = nil **then return**
3. **if** left[x]=nil or right[x]=nil
4. **then** z ← x
5. **else** z ← Successor(x,T)

procedure SpliceOut(z) O(1) time
(* Exercise *)
remove node z and
bypass link between p[z] and
lone child of z (maybe nil too)
end

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder







Running Time:
O(length of search path of z) = O(h).

BST Height h

Search	Minimum	Predecessor
Insert	Maximum	Successor
Delete		

- All these path following routines take at most $O(h)$ time.

Assignment Project Exam Help

- $\lfloor \log n \rfloor \le h < n$.

https://powcoder.com

- h could be as bad as $\Theta(n)$ if the tree is extremely unbalanced.

Add WeChat powcoder

- To improve, we will study search trees that are efficient in the
  - worst-case sense: Red-Black trees, B-trees, 2-3-4 trees.
  - amortized sense: Splay trees.

these are multi-way search trees

Multi-way Search Trees

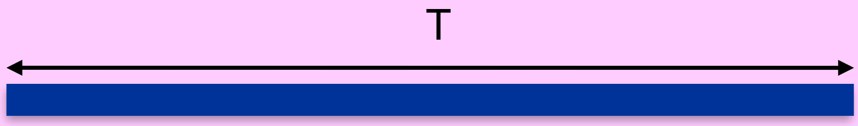
Assignment Project Exam Help

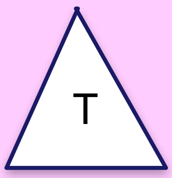
https://powcoder.com

Add WeChat powcoder

Split: Multi-Way vs Binary

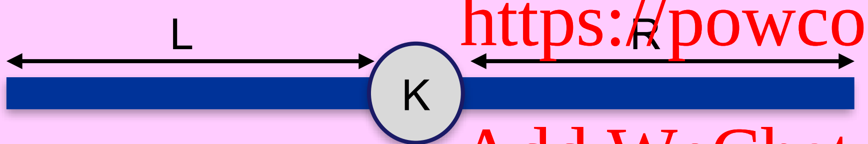
Sorted key sequence □ **Inorder key sequence**



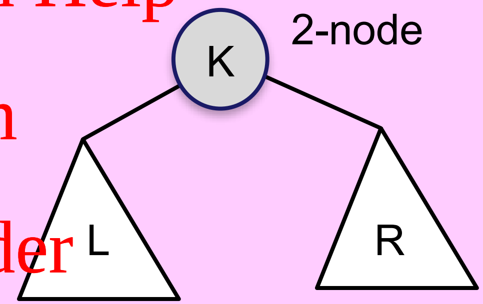


Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

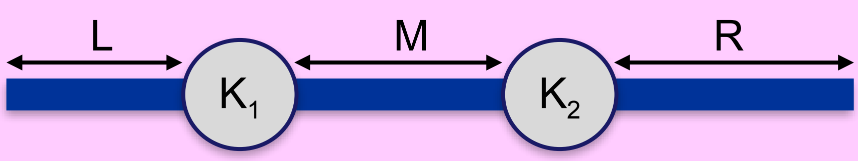
binary split:



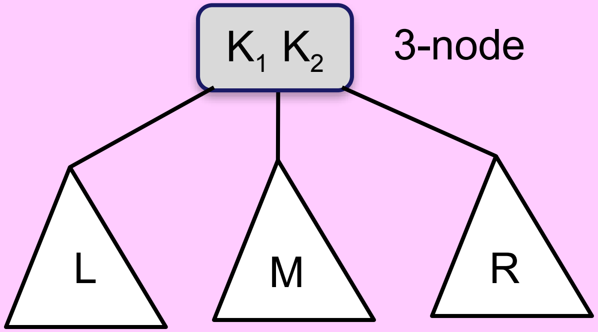
($\forall$ key in L) < K < ($\forall$ key in R)



ternary split:

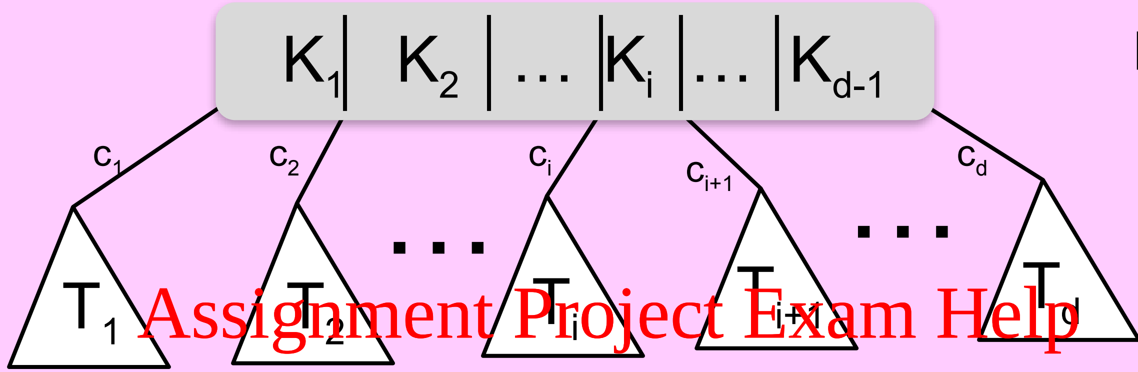


($\forall$ key in L) < K_1 < ($\forall$ key in M) < K_2 < ($\forall$ key in R)



Multi-Way Search Tree

root x (a d-node)



$K_i = Key_i[x]$
$c_i = c_i[x]$

Assignment Project Exam Help

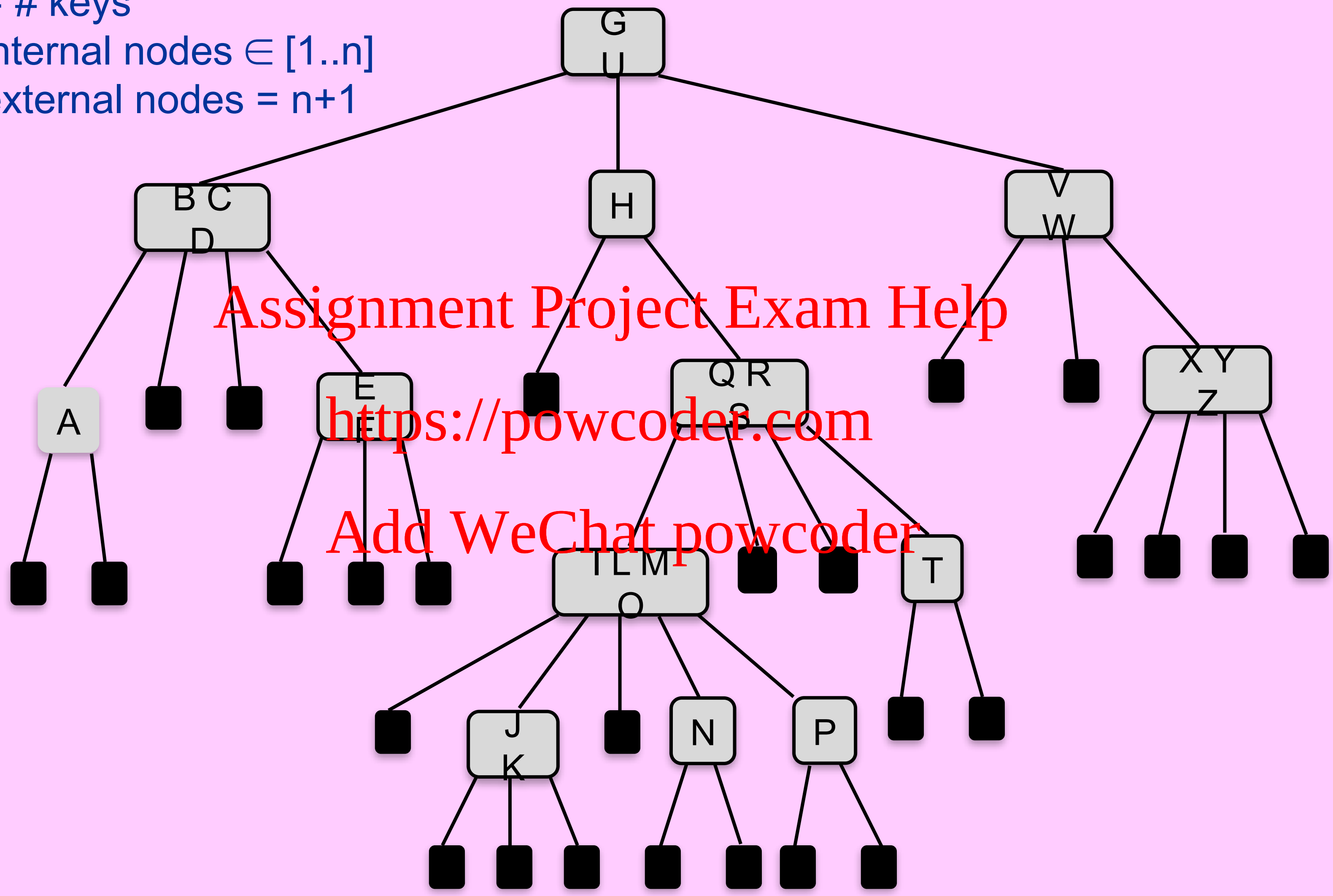
https://powcoder.com

Add WeChat powcoder

1. Root is a d-node for some $d \geq 2$.
2. $K_1 < K_2 < \ldots < K_i < \ldots < K_{d-1}$.
3. (every key in T_i) < K_i < (every key in T_{i+1}), for i = 1..d-1.
   (3 implies 2.)
4. Each subtree T_i , i=1..d, is a multi-way search tree.
5. The empty tree is also a multi-way search tree.

Example

n = # keys
# internal nodes ∈ [1..n]
# external nodes = n+1



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Exercises

https://powcoder.com

Add WeChat powcoder

1. **[CLRS, Exercise 12.2-1, page 293]** Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequences could **not** be the sequence of nodes examined? Explain.
   - (a) 2, 252, 401, 398, 330, 344, 397, 363.
   - (b) 924, 220, 911, 244, 898, 258, 362, 363.
   - (c) 925, 202, 911, 240, 912, 245, 363.
   - (d) 2, 399, 387, 219, 266, 382, 381, 278, 363.
   - (e) 935, 278, 347, 621, 299, 392, 358, 363.

2. **[CLRS, Exercise 12.2-4, page 293]** Suppose the search path for a key K on a BST ends up in an external node. Let A be the set of keys to the left of the search path; B be the set of keys on the search path; and C be the set of keys to the right of the search path. Give a smallest counter-example to refute the claim that ∀a∈A, ∀b∈B, ∀c∈C, we must have a ≤ b ≤ c.

Assignment Project Exam Help

https://powcoder.com

3. **[CLRS, Exercise 12.3-4, page 299]** Is the Delete operation on BST "commutative" in the sense that deleting x and then y from the BST leaves the same tree as deleting y and then x? Argue why it is or give a counter-example.

Add WeChat powcoder

4. **[CLRS, Exercise 12.2-8, page 294]** Give a proof by the **potential function method** for the following fact: No matter what node x we start at in an arbitrary height h BST T, R successive calls to Successor, as shown below

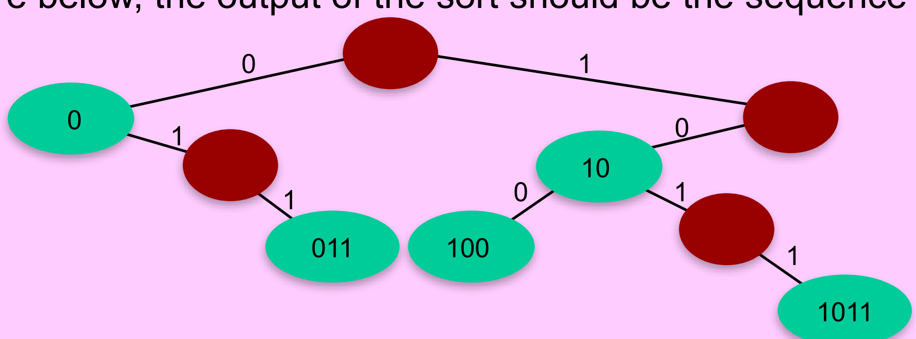
   for i ← 1..R **do** x ← Successor(x,T)

   takes at most O(h+R) time. [Note: O(h·R) is obvious.] Carefully define the potential function.

5. **Range-Search Reporting in BST:** Let T be a given BST. We are also given a pair of key values a and b, a < b (not necessarily in T). We want to report every item x in T such that a ≤ key[x] ≤ b. Design an algorithm that solves the problem and takes O(h+R) time in the worst case, where h is the height of T and R is the number of reported items (i.e., the output size). Prove the correctness of your algorithm and the claimed time complexity.
   [Hint: there is a short and elegant recursive solution.]

6. **Binary Tree Reconstruction:** Which of the following pairs of traversal sequences uniquely determine the Binary Tree structure? Fully justify each case.
(a) Preorder and Postorder.
(b) Preorder and Inorder.
(c) Levelorder and Inorder.

7. **[CLRS, Problem 12-2, page 304] Radix Trees:**
Given two strings $a = a_0 a_1 \ldots a_p$ and $b = b_0 b_1 \ldots b_q$, where each a_i and each b_j is in some ordered set of characters, we say that string a is **lexicographically less than** string b if either
(i) $\exists$ an integer j, where $0 \le j \le \min\{p,q\}$, such that $a_i = b_i$ $\forall i=0,1,\ldots,j-1$, and $a_j < b_j$, or
(ii) $p < q$ and $a_i = b_i$ for all $i=0,1,\ldots,p$.
Assignment Project Exam Help
For example, if a and b are bit strings, then 101011 < 10110 by rule (i) (j=3) and 10100 < 101000 by rule (ii). This is similar to the ordering used in English-language dictionaries.
The radix tree data structure shown below stores the bit strings 1011, 10, 011, 100, and 0.
https://powcoder.com
When searching for a key $a = a_0 a_1 \ldots a_p$, we go left at a node of depth i if $a_i = 0$ and right if $a_i = 1$. Note that the tree uses some extra "empty" nodes (the dark ones).
Let S be a set of distinct binary strings given in some arbitrary unsorted order, whose string lengths sum to n.
Add WeChat powcoder
(a) Show an O(n) time algorithm to construct a radix tree with O(n) nodes that stores the strings in S.
(b) Show how to use the radix tree just constructed to sort S lexicographically in O(n) time. In the figure below, the output of the sort should be the sequence 0, 011,10,100,1011.



8. **Iterative Inorder:** We gave a linear-time non-recursive Inorder procedure using parent pointers.
(a) If parent pointers are not available, one could maintain a stack holding the ancestors of the current node. Write such a procedure and analyze its running time.
(b) Write a linear-time non-recursive in-place Inorder procedure without parent pointers. (In-place means you cannot use any stack or equivalent; just the given tree and O(1) additional scalar variables.) [Hint: temporarily modify the tree links then put them back into their original form.]

9. **BST construction lower bound:** We are given a set S of n keys and want to algorithmically construct a BST that stores these keys.
(a) Show that if the keys in S are given in sorted order, then there is an O(n) time solution.
Assignment Project Exam Help
(b) Show that if the keys in S are given in arbitrary order, then any off-line algorithm that solves the problem must, in the worst-case, take at least Ω(n log n) time in the decision tree model of computation. [Note: there are algorithms that do not sort S as a first step!]
https://powcoder.com

Add WeChat powcoder
10. **Split and Join on BST:** These are cut and paste operations on dictionaries. The Split operation takes as input a dictionary (a set of keys) A and a key value K (not necessarily in A), and splits A into two disjoint dictionaries B = { x∈A | key[x] ≤ K } and C = { x∈A | key[x] > K }. (Dictionary A is destroyed as a result of this operation.) The Join operation is essentially the reverse; it takes two input dictionaries A and B such that every key in A < every key in B, and replaces them with their union dictionary C = A∪B. (A and B are destroyed as a result of this operation.)
Design and analyze efficient Split and Join on binary search trees.
[Note: there is a naïve slow solution for Split (similarly for Join) that deletes items from A one at a time and inserts them in B or C as appropriate. Can you do it more efficiently?]

11. **Multi-way Search Tree Traversals:** Given a multi-way search tree with n keys, give O(n) time algorithms to print its keys in Preorder, Postorder, Inorder, Levelorder.

12. **Multi-way Search Tree Search:** Given a multi-way search tree T and a key K, describe how to

END

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder