

230. Kth Smallest Element in a BST [↗ \(/problems/kth-smallest-element-in-a-bst/\)](/problems/kth-smallest-element-in-a-bst/)

April 30, 2019 | 82.3K views

Average Rating: 4.61 (70 votes)

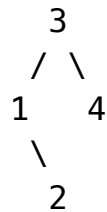
Given a binary search tree, write a function `kthSmallest` to find the **k**th smallest element in it.

Note:

You may assume `k` is always valid, $1 \leq k \leq$ BST's total elements.

Example 1:

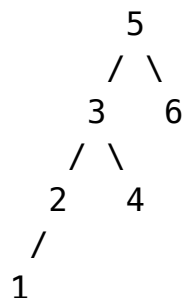
Input: `root = [3,1,4,null,2]`, `k = 1`



Output: 1

Example 2:

Input: root = [5,3,6,2,4,null,null,1], k = 3



Output: 3

ô”ỀT óHh

Y Û m̄n̄X^gK ă h̄” l̄ ă ă H̄kX̄m̄ X̄X̄” H̄X̄H̄ă ẹk̄” N̄X̄ ẹl ấ” ó ẹXl m̄ N̄l n̄X̄ ău k̄h̄ Ềk̄m̄H̄ óX̄m̄ đ
 §” T̄ T̄” óỀ ấ” ó” H̄ă ăX̄ n̄X̄ ău ḡh̄ Ềk̄m̄H̄ ón̄ ăX̄

g” Ềă ẹ

§” T̄ m̄ n̄l t̄ X̄H̄X̄ n̄X̄ n̄X̄

K̄X̄H̄ H̄k̄m̄” n̄X̄X̄H̄ Ềk̄m̄ n̄X̄ m̄ n̄l t̄ X̄H̄X̄ n̄X̄H̄

- Z̄đđ ă ă Ề đ̄ ề d̄HDFS t̄

Ền̄ă k̄m̄H̄ n̄j̄T̄X̄ l̄” H̄m̄X̄ depth k̄ n̄X̄H̄ă ău k̄” n̄ m̄ ẹXT̄” óỀ k̄m̄H̄ n̄ ầ” m̄ ẹl H̄X̄ l̄ Ền̄X̄
 T̄ ấ” T̄ ẹ m̄ \X̄m̄ Ề N̄l n̄X̄ 6 ăm̄ H̄” m̄ H̄X̄ l̄ ẹ” n̄X̄H̄ ẹl ầ

Để tìm kiếm phần tử nhỏ thứ k trong cây nhị phân tìm kiếm (BST), chúng ta cần duyệt qua các phần tử theo thứ tự tăng dần. Có ba cách để duyệt qua các phần tử trong BST: preorder, inorder và postorder. Trong đó, inorder là cách duyệt qua các phần tử theo thứ tự tăng dần.

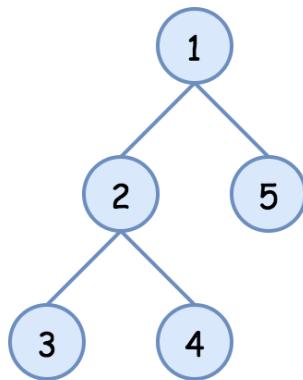
- Nếu k là 1, chúng ta cần duyệt qua các phần tử theo thứ tự tăng dần.

Để tìm kiếm phần tử nhỏ thứ k trong cây nhị phân tìm kiếm (BST), chúng ta cần duyệt qua các phần tử theo thứ tự tăng dần. Có ba cách để duyệt qua các phần tử trong BST: preorder, inorder và postorder. Trong đó, inorder là cách duyệt qua các phần tử theo thứ tự tăng dần.

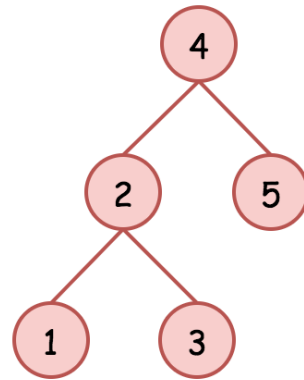
Để tìm kiếm phần tử nhỏ thứ k trong cây nhị phân tìm kiếm (BST), chúng ta cần duyệt qua các phần tử theo thứ tự tăng dần. Có ba cách để duyệt qua các phần tử trong BST: preorder, inorder và postorder. Trong đó, inorder là cách duyệt qua các phần tử theo thứ tự tăng dần.

DFS Preorder: Node -> Left -> Right
DFS Inorder: Left -> Node -> Right
DFS Postorder: Left -> Right -> Node
BFS: Node -> Left -> Right

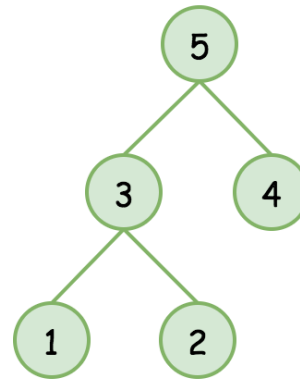
Traversal = [1, 2, 3, 4, 5]



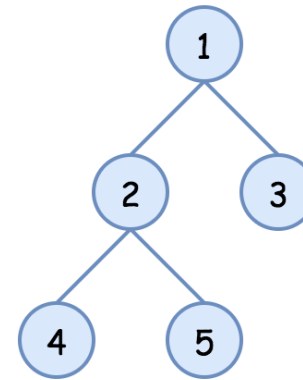
[root.val] +
preorder(root.left) +
preorder(root.right)
if root else []



inorder(root.left) +
[root.val] +
inorder(root.right)
if root else []



postorder(root.left) +
postorder(root.right) +
[root.val]
if root else []



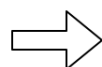
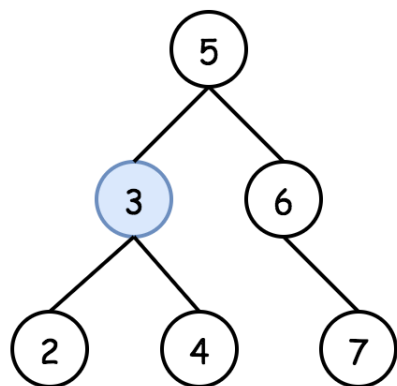
iterations
with the queue

Để tìm kiếm phần tử nhỏ thứ k trong cây nhị phân tìm kiếm (BST), chúng ta cần duyệt qua các phần tử theo thứ tự tăng dần. Có ba cách để duyệt qua các phần tử trong BST: preorder, inorder và postorder. Trong đó, inorder là cách duyệt qua các phần tử theo thứ tự tăng dần.

2nd smallest element = ?

For a given BST, the inorder traversal gives the elements in sorted order. The 2nd smallest element is the 2nd element in the sorted array.

2th smallest element = ?



[2, 3, 4, 5, 6, 7]



2th smallest element = 2th element in the inorder traversal

Java

Python

 Copy

```

1 class Solution {
2     public ArrayList<Integer> inorder(TreeNode root, ArrayList<Integer> arr) {
3         if (root == null) return arr;
4         inorder(root.left, arr);
5         arr.add(root.val);
6         inorder(root.right, arr);
7         return arr;
8     }
9
10    public int kthSmallest(TreeNode root, int k) {
11        ArrayList<Integer> nums = inorder(root, new ArrayList<Integer>());
12        return nums.get(k - 1);
13    }
14 }

```

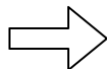
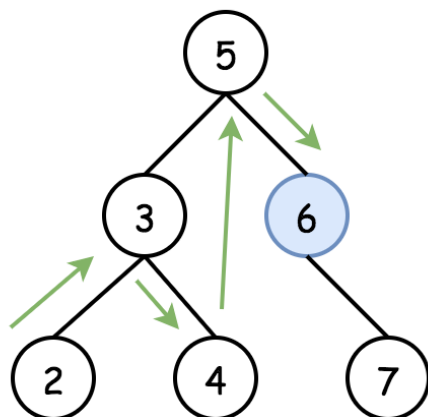
b" h HXUā 2e Eāk

- Kā Xl" h HXUā $O(N)$ m 6ōā rñt Xk ĩ
- gH lXl" h HXUā $O(N)$ m āXH e ā" H Xñt Xk ĩ

2H.H lÜ 'HāHā e

KUX 6" t Xñōkā e l" óE 6Xl" e Xñ ām āHā eñT āñUX ĪE" NñlāKÜ T ũ" eXl" óE kHXI óHñX k" Eā e 6Xl ókXñXā e" eXI m 6ōā nUX XēāXā" H Xñt Xk ĩ e l" eXl" óE kñH NññX āñXēñ Xēñ

5th smallest element = ?



stack = [~~5~~, ~~3~~, ~~2~~, ~~4~~, ~~6~~]

5th smallest element = 6

Java

Python

Copy

```

1 class Solution {
2     public int kthSmallest(TreeNode root, int k) {
3         LinkedList<TreeNode> stack = new LinkedList<TreeNode>();
4
5         while (true) {
6             while (root != null) {
7                 stack.add(root);
8                 root = root.left;
9             }
10            root = stack.removeLast();
11            if (--k == 0) return root.val;
12            root = root.right;
13        }
14    }
15 }

```

b" h H 2 e E k a

- [illegible]

$$\hat{O}'' \hat{E} \hat{T} \hat{O} \hat{H}$$

Ý Û m̃āñx̃^g_kā_h̃"l̃āñ h̃ēk_x̃h̃ñx̃ēk_x̃"̃h̃h̃r̃ā_ēk̃"̃ñl̃ē_ēl̃_ũ"̃ō_ē_xl̃ m̃ñl̃ r̃ū_x̃āñk̃h̃ ēk̃m̃
 ñl̃h̃ó_x̃ēñd̃s̃"̃t̃t̃"̃ōē_ũ"̃ō"̃h̃āñā_x̃r̃ū_x̃āñg̃h̃ ēk̃m̃h̃ōñē_x̃d̃

[illegible]

§ Xe\XT^{ab}” om eũ” H^hñ ã rã e ã₃X₃ññ X₃X₃“l₃X₃ H₃U” N₃ñ X₃ñ X₃m₃U k O(2H + k) \” h H₃X₃ññ” T m
 ” H^hñ ã X₃n₃ñ

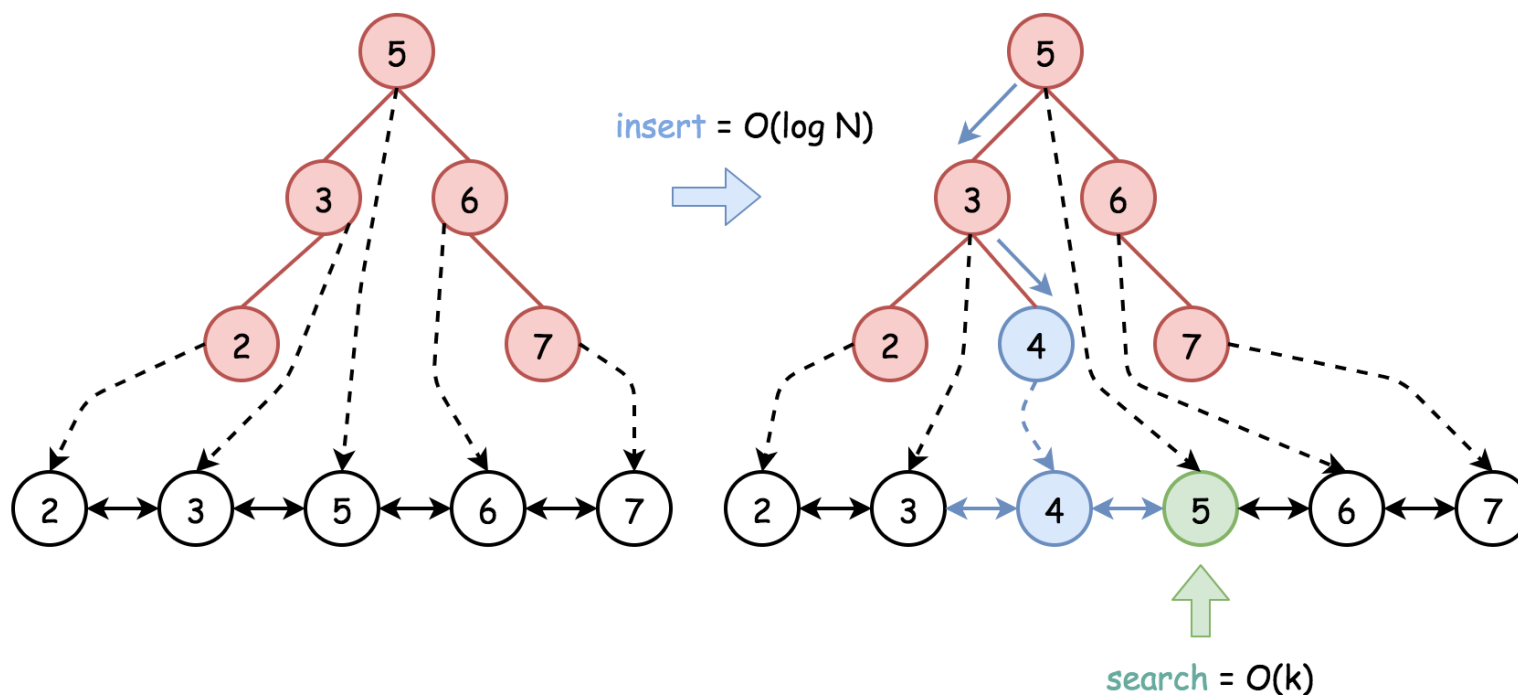
Kŭ n̄k | Xkəj hōXkrā ɛ̃n̄ kǎ 𑜀𑜂𑜆𑜨 T X̣j kəʔ m̄ ẵ Hē̃ Xəm k̄r̄l̄ n̄l̄ XT Ū̃ Ū̃” ɛ̃m̄ t̄ək ^gK t̄ək̄ X ɛ̃l
 ”H̄ẵ t̄Xk r̄l̄X N̄l̄ T t̄ən̄j” H̄X H̄ẵ t̄ək ‘H

- $\text{ek}^{\text{h}}\text{X}^{\text{h}}\text{in}$
- $\text{k}^{\text{h}}\text{X}^{\text{h}}\text{X}$
- $\text{ô}^{\text{h}}\text{el}^{\text{h}}\text{a}^{\text{h}}\text{kh}^{\text{h}}\text{X}^{\text{h}}\text{m}$

gốl ǔ k_hǎl mǎk ʔ ốỀ h_h ǎ xh

- $\mathcal{O}(H)$ rấ X Nũ Hũ X ặ k X h n e l | X ẽ n xĩ
- $\mathcal{O}(k)$ Nũ Hũ X k X Ĥ Ū” N ẽ u k h Ì ẽ k mĩ

insert 4
and then search for the 4th smallest


$$K\bar{U}X" \uparrow X_H \bar{E} \bar{a} X \setminus " \check{h} H \bar{E} \bar{U} \bar{a} N \bar{J} \bar{H} \bar{e} k X \bar{H} \bar{I} X \bar{E} X \setminus " \downarrow k X \bar{H} \bar{U} " N \bar{a} \bar{b} k \check{h} \bar{E} \bar{X} k \bar{a} k O(H + k) \bar{e} k \bar{a} X \downarrow " N O(2H + k) \hat{\eta}$$

b”h HẸXƯẦỮ 2ẹ Ìikạk

- 2e Ēkāk TĥmXe 6ü ? Ēāk" e HĭmK'NĔXm" | Xŋ" ħ Ēāk" eI el ? el t Hĭ HĭmK'NĔXm" | Xŋ" ħ 1 el t Hĭ

È n̄X n̄ūā H̄ā Ē' H

!ÔXỨn H Hã Ẽk1" ẹknó\m6knHj h ấHX" H Xứn t Xk Ẽ

g" Hm^ ü ▼

👁 !ĒHt āT

 $\bar{E}'' \text{ km}$

k̄ũk̄ũáǎēΣΛη k̄ũk̄ũáǎēΣΛι ★! 5. ①! ÷óĤ ħ. SM "H Êĭ

77 ^ v | ↻ gÜ ㅈ | ↶ EXH

$$\bar{U} \, H \bar{U} \bar{X} \bar{W} \, " \, H \bar{U} \, H \bar{U} \bar{X} \bar{W} \, " \, I \, \star! \, \ominus! \, \hat{u} \, _ \, \hat{n} \, _ \, \text{SM} \, _ \, \text{SM} \, 2 \hat{1}$$

ô”HX [Ü̈]e”I Xä nũX mĩKŋTjHX[ŋ]”ñ HõnX Ü̈T ħ eũ Hk m nũX Ænd el Ü̈T ħ eũ Hk m nũX HũmĩKÜ̈Xe mX [Ü̈]e”I XŋTjX T äæ”T T Ü̈nũX HũX ænũ kñ Ækmak m nũX Ænd HũX HũmĩHẽUĩngX HũñækX Hñĩ XÆX H K È Hİ İn

^ v ↻ gÜ ½ ↶ EXH

g§ · Ŷ 7 Èt Ē±AĒg

gſ' Ŷ ƆĖt Ē†AĖg

· n̄X³H³ā³X³H³ ” t³X³m̄³H³n̄³Ē³n̄³Y ” H³X³H³m̄³e³n̄³æ³X³h̄³m̄³X³e³X³ e ” l³X³Ē³n̄³Y

EX I Î " HX

$$g\delta \cdot \hat{Y} \quad \dot{E}t \bar{E} \pm A \dot{E}g$$

10/12

EX I Î " H

EX I Î " HЖ

11/12



b" HÜJÜm̄Ω = 9 = 9 ‡XXrb" I X
§ XĒb XeṁX̄HkōH' Hm̄ lḡKX̄h̄ k̄ HṁX̄h̄ k̄H lḡ ĒH̄ ṽ Ē" Ǣ ũ H̄H̄ ṽH̄
🇺🇸 ṽēX̄ ḡm̄X̄k̄ H̄X̄H̄ ḡH̄