

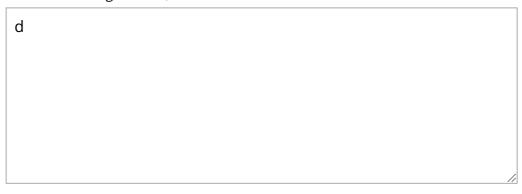
1/1 points

1.

**Java autoboxing and equals()**. Consider two **double** values  $\mathbf{a}$  and  $\mathbf{b}$  and their corresponding <tt>Double</tt> values  $\mathbf{x}$  and  $\mathbf{y}$ .

- Find values such that (a == b) is true but x. equals(y) is false.
- Find values such that (a == b) is false but x. equals(y) is true.

Note: these interview questions are ungraded and purely for your own enrichment. To get a hint, submit a solution.



Your answer cannot be more than 10000 characters.

## Thank you for your response.

Hint: IEEE floating point arithmetic has some peculiar rules for 0.0, -0.0, and NaN. Java requires that equals () implements an equivalence relation.



1/1 points

2.

**Check if a binary tree is a BST.** Given a binary tree where each **Node** contains a key, determine whether it is a binary search tree. Use extra space proportional to the height of the tree.

d //

Your answer cannot be more than 10000 characters.

## Thank you for your response.

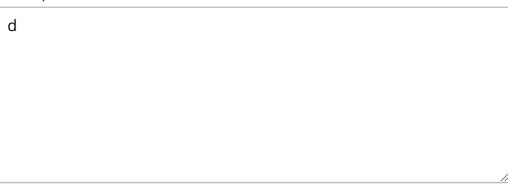
Hint: design a recursive function isBST(Nodex, Keymin, Keymax) that determines whether x is the root of a binary search tree with all keys between min and max.



1/1 points

3.

**Inorder traversal with constant extra space**. Design an algorithm to perform an inorder traversal of a binary search tree using only a constant amount of extra space.



Your answer cannot be more than 10000 characters.

## Thank you for your response.

**Hint:** you may modify the BST during the traversal provided you restore it upon completion.



1/1 points

1

**Web tracking.** Suppose that you are tracking n web sites and m users and you want to support the following API:

- User visits a website.
- How many times has a given user visited a given site?

What data structure or data structures would you use?

d		

Your answer cannot be more than 10000 characters.

## Thank you for your response.

*Hint*: maintain a symbol table of symbol tables.