Interview Questions: Balanced Search Trees (ungraded)

Practice Quiz, 3 questions

Congratulations! You passed!

Next Item

1 / 1 point

1. **Red–black BST with no extra memory.** Describe how to save the memory for storing the color information when implementing a red–black BST.

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

Save extra memory for storing the color information:

Modify the structure of the BST to encode the color information.

Your answer cannot be more than 10000 characters.

Thank you for your response.

Hint: modify the structure of the BST to encode the color information.

---

1 / 1 point

2. **Document search.** Design an algorithm that takes a sequence of n document words and a sequence of m query words and find the shortest interval in which the m query words appear in the document in the order given. The length of an interval is the number of words in that interval.

n document words and m query words.

for each word, maintain a sorted list of the indices in the document in which that word appears. Scan through the sorted lists of the query words in a judicious manner.

Your answer cannot be more than 10000 characters.

Thank you for your response.

Hint: for each word, maintain a sorted list of the indices in the document in which that word appears. Scan through the sorted lists of the query words in a judicious manner.

---

1 / 1 point

3. **Generalized queue**. Design a generalized queue data type that supports all of the following operations in logarithmic time (or better) in the worst case.

- Create an empty data structure.
- Append an item to the end of the queue.
- Remove an item from the front of the queue.
- Return the i^{th} item in the queue.
- Remove the i^{th} item from the queue.

Generalized Queue.

Created a red-black BST where the keys are integers and the values are the items, such that the ith largest integer key in the red-black BST corresponds to the ith item in the queue.

Your answer cannot be more than 10000 characters.

Thank you for your response.

Hint: create a red–black BST where the keys are integers and the values are the items such that the i^{th} largest integer key in the red–black BST corresponds to the i^{th} item in the queue.