Algorithms Example Sheet 3: Problems

Binary Search Trees (BSTs)

Exercise 3.P.1 [More core BST operations]

(a) Write pseudocode for each of the following operations, argue about their correctness and state their time complexity:

  i. Finding the pre-order/in-order/post-order traversal of a BST.

  ii. Finding the level-order traversal of a BST.

  iii. Finding the height of a BST.

  iv. Finding the number of leaves in a BST.

  v. Finding the total number of nodes in a BST.

  vi. Attempt **[2009P1Q5 (e)]**.

(b) Your friend decides to implement the inorder traversal by finding the successor n times. What is the time complexity of this approach?

Exercise 3.P.2 Professor Bunyan thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path. Professor Bunyan claims that any three keys $a \in A$, $b \in B$, and $c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim.

Exercise 3.P.3 Given n integers $x_1, \ldots, x_n$ and an integer k, determine if there are two elements i_1 and i_2 such that $x_{i_1} + x_{i_2} = k$ (using a binary search tree). What is the time complexity of your approach?

Exercise 3.P.4 [Concert Tickets] There are n concert tickets available, each with a certain price. Then, m customers arrive, one after another. Each customer announces the maximum price he or she is willing to pay for a ticket, and after this, they will get a ticket with the nearest possible price such that it does not exceed the maximum price.

Design an algorithm that given the sequence of customers, determines the price that each customer will pay for their ticket or that no such sequence exists.

Source: **CSES 1091**

Exercise 3.P.5 [Josephus problem] Consider a game where there are n children (numbered $1, 2, \ldots, n$) in a circle. During the game, repeatedly k children are skipped and one child is removed from the circle. In which order will the children be removed?

Source: **CSES 2163**

Exercise 3.P.6 We are throwing points on the x-axis and we are interested to know after each throw which two neighbouring points are farthest away. For example, if we throw $x_1 = 1, x_2 = 8, x_3 = 3, x_4 = 5$. After x_2 the distance is 7, after x_3 the distance is 5, after x_4 the distance is 3. How can you achieve this efficiently?

2-3-4 trees / Red-Black Trees

Exercise 3.P.7 Explain how to modify a RBT so that we are able to find the k-th larget element efficiently. (Note: k is not fixed)

Exercise 3.P.8 Attempt **[2009P1Q5 (a)-(d)]**.

Exercise 3.P.9 [Rotations]

(a) Define *left* and *right* rotations. Give an example for each.

(b) Argue that these preserve the BST property.

(c) Show that any arbitrary n-node binary search tree can be transformed into any other arbitrary n-node binary search tree using $\mathcal{O}(n)$ rotations. (**Hint:** *First show that at most* $n-1$ *right rotations suffice to transform the tree into a right-going chain.*)

(d) Show how rotations are used to implement insertion in RBTs.

(e) Attempt **[2007P1Q12 (d)]**.

(f) Attempt **[2017P1Q7]**.

(g) (+) Attempt **[2014P1Q8]**.

Exercise 3.P.10 Attempt **[2018P1Q8]**.

1 B-trees

Exercise 3.P.11 Attempt **[2007P11Q9]**.

Hash tables

Exercise 3.P.12 Implement and compare the three hashing approaches on a set of randomly generated values.

Exercise 3.P.13 Attempt **[2008P10Q9]**.

Exercise 3.P.14 Attempt **[2017P1Q8]**.

Exercise 3.P.15 How would you use a hash table to create a basic spell checker (i.e. decide whether or not a word is correctly spelt)? For an incorrectly-spelt word, how would you efficiently provide a set of correctly-spelt alternatives? (optional) Look at trie data structures for making this more efficient.

[Source: Robert Harle]

Priority queues

Exercise 3.P.16 [Merging priority queues]

(a) Explain how you could merge two priority queues of n and m items respectively by making calls to `extractMin()` and `insert()`. How many operations do you need?

(b) What is the time complexity of your approach for min-heap? How can you speed it up?

Exercise 3.P.17 Attempt **[2008P1Q4]**.

Exercise 3.P.18 You are given n integers $x_1, \ldots, x_n$ one by one and an integer $k \in [n]$. Using two priority queues, for each $i \geq k$, report the k-th largest element of $x_1, \ldots x_i$. What is the time complexity of your approach?

Binary heaps

Exercise 3.P.19 [Binomial heap - Implementation] Implement the binomial heap. You may find helpful p.20-25 in Okasaki's "Purely functional data structures" book.

Exercise 3.P.20 [Binomial heaps - efficient initialisation] Describe an algorithm to initialise a binomial heap for n items in $\mathcal{O}(n)$ time.

Exercise 3.P.21 [Binomial heaps - small variants]

(a) Explain how you can modify `first()` to run in $\mathcal{O}(1)$ time.

(b) Explain how you can modify `insert()` to run in amortised $\mathcal{O}(1)$ time (You will need to use the efficient initialisation and also modify `extractMin()`).

Exercise 3.P.22 Attempt **[1996P8Q6]**.

Exercise 3.P.23 [Binomial heaps - add-to-all] (+) Explain how you would implement the `add-to-all(v)` method, which adds v to all elements currently in the binomial heap. Make sure that all operations are still working.