

CSC 225 (Fall 2016): Algorithms and Data Structures I

Assignment 3—theoretical part

Note: your solution must be submitted as a **single pdf file**

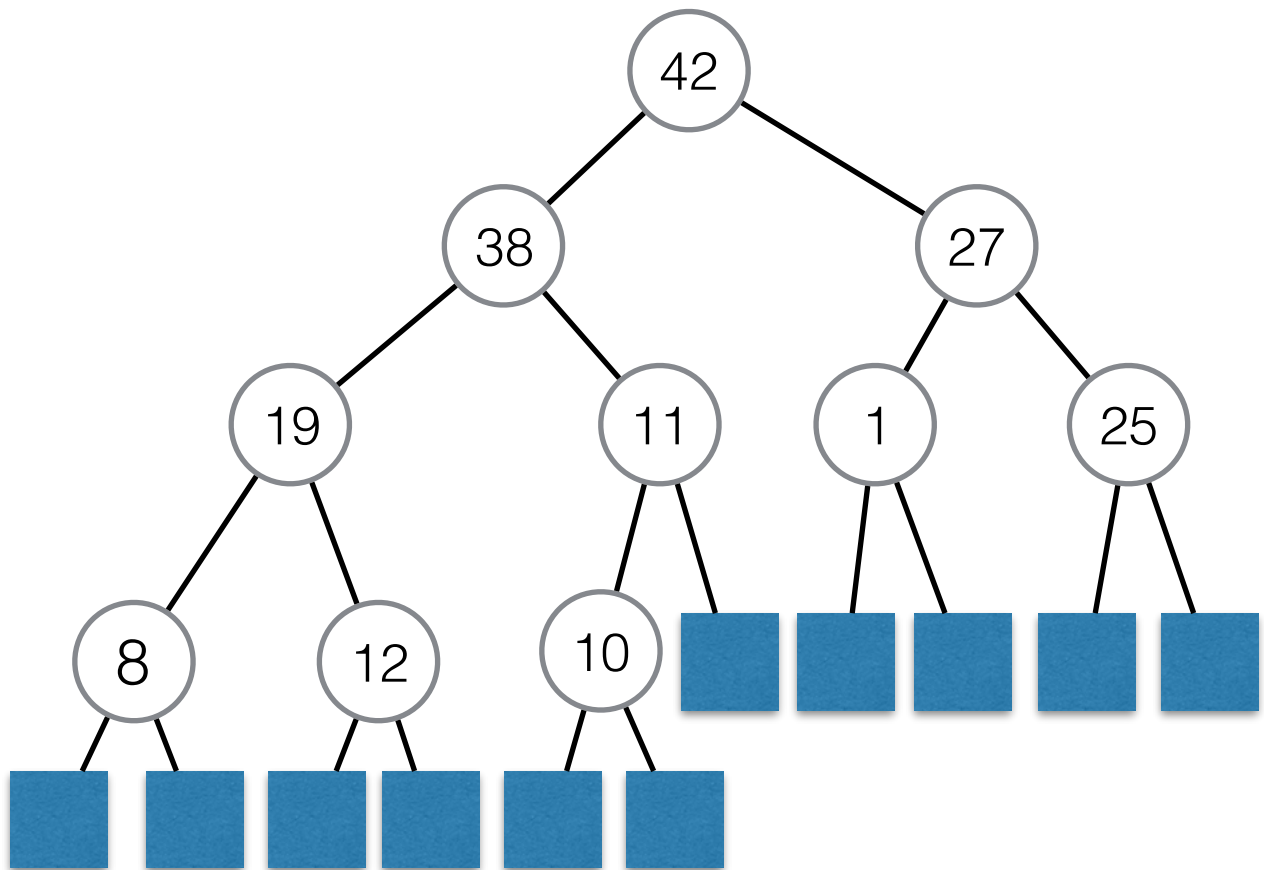
Question 1. You are given a collection of n bolts of different widths and n corresponding nuts. You are allowed to try a nut and bolt together, from which you can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together, or two bolts together. The goal is to match each bolt to its nut.



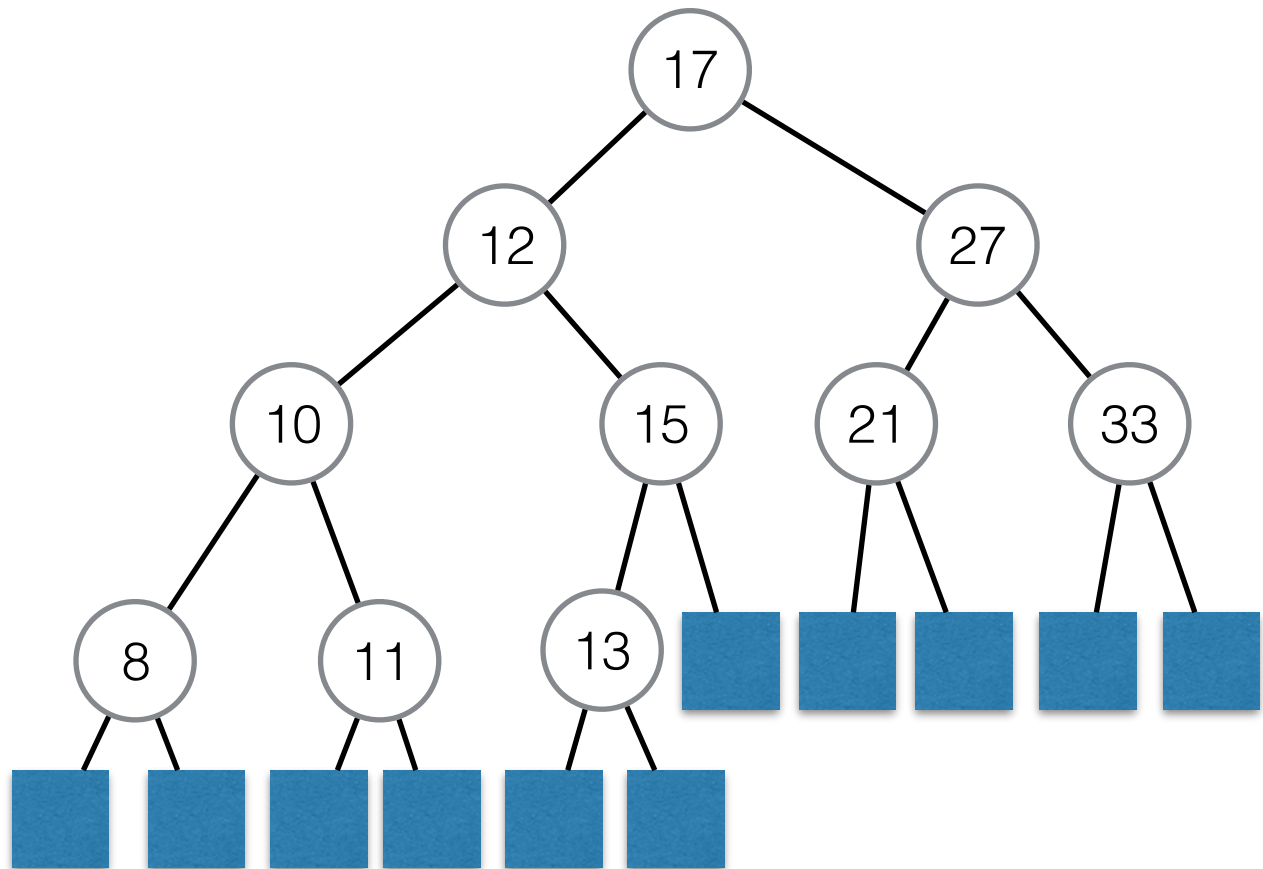
In pseudocode, design an algorithm for this problem with an *expected* running time of $O(n \log n)$.

Question 2.

- a) Consider the following heap. Draw the heap that would result after deleting the maximum element. Recommendation: show your work.



b) Consider the following binary search tree. Draw the binary search tree that would result after deleting the element with key 17. Recommendation: show your work.



Question 3. If we insert a set of n items into a binary search tree using the $\text{insert}(k, e)$ method as discussed in class, the resulting tree may possibly have a height of $O(n)$. However, if instead we have a randomly built binary search tree the expected height is $O(\lg n)$. To be more exact, a *randomly built binary search tree* on n keys is one that arises from inserting the keys in random order into an initially empty tree, where each of the $n!$ permutations of the input keys is equally likely.

Thus, to build an expected balanced tree for a fixed set of items given in advance, we could randomly permute the items and then insert them in that order into an originally empty tree.

A problem arises if we are not given all the items in advance of building the tree.

To accommodate this nice property in the case that not all elements are given ahead of time, we define the data structure, called *treap*, that can be considered a combination of a binary search tree and a (min)-heap. One can show that the expected height of a treap is $O(\lg n)$, and hence the expected time to search for a value in the treap is $O(\lg n)$.

More exactly, a **treap** is defined as a binary tree in which every node has both a *search key* and a *priority*, where:

- (1) The inorder-sequence of search keys is sorted. That is, if node v is a left child of u , then $key[v] < key[u]$; if v is a right child of u , then $key[v] > key[u]$ (a property of *binary search tree*),
- (2) Each node's priority is *smaller* than the priorities of its children. That is, if v is a child of u , then $priority(v) > priority(u)$ (a property of *min-heap*).

Argue convincingly:

- 1) A treap is exactly the binary search tree that results by: inserting the nodes, one at a time, into an initially empty tree, in order of increasing priority—using the usual binary search tree insertion algorithm.
- 2) Explain how treap-insert works. Explain the idea first in English and then give pseudocode. (*Hint: Execute the usual binary search tree insert and then perform rotations to restore the min-heap order property.*)
- 3) What is the height of a treap with n elements in the worst-case? Argue convincingly.

Question 4. A *halloween number* is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are *halloween numbers*.



An example of a halloween number is 23:

- $2^2 + 3^2 = 13$
- $1^2 + 3^2 = 10$
- $1^2 + 0^2 = 1$

22 is not a halloween number as the process will never reach 1:

- $2^2 + 2^2 = 8$
- $8^2 = 64$
- $6^2 + 4^2 = 52$
- $5^2 + 2^2 = 29$
- $2^2 + 9^2 = 85$
- $8^2 + 5^2 = 89$
- $8^2 + 9^2 = 145$

- $1^2+4^2+5^2 = 42$
- $4^2+2^2 = 20$
- $2^2+0^2 = 4$
- $4^2 = 16$
- $1^2+6^2 = 37$
- $3^2+7^2 = 58$
- $5^2+8^2 = \mathbf{89}$
- ...

In pseudocode and using the concept of hashing, describe an efficient algorithm to determine whether or not a number is a halloween number.