Quiz on basic data structures

F. Morain

September 8, 2020

The goal is to help you evaluate your familiarity with basic topics in programming. We won't give you a mark for this. If everything looks fine, then probably, you don't need to attend this refresher.

A01 Let `t` be an array of n integers. What is the complexity of finding a given integer in `t`?

1. $O(n)$
2. $O(1)$
3. $O(\log n)$
4. $O(n^2)$

A02 Let `t` be a **sorted** array of n integers. What is the complexity of finding a given integer in `t`?

1. $O(n)$
2. $O(1)$
3. $O(\log n)$
4. $O(n^2)$

A03 Let `t` be an array of n integers. What is the best complexity of sorting `t` using comparisons of integers?

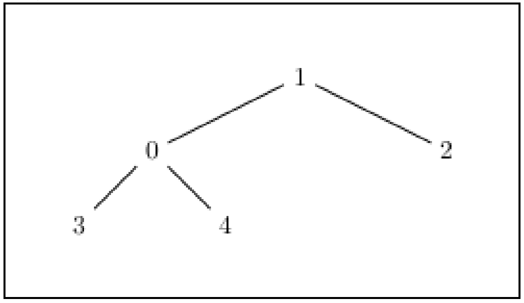
1. $O(n)$
2. $O(1)$
3. $O(n \log n)$
4. $O(n^2)$

A04 Let `t` be an array of n integers. Which algorithm has a good average complexity?

1. quicksort
2. bubble sort
3. insertion sort

4. selection sort

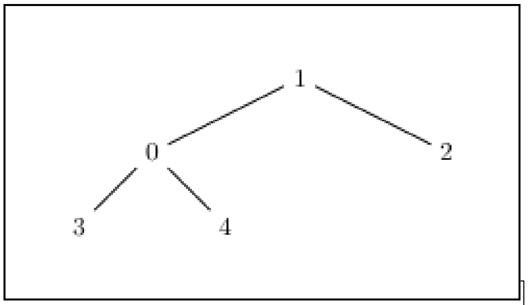
T01 In this picture, what are the nodes of the tree?



1. 0, 1, 2, 3, 4
2. 1
3. 2, 3, 4
4. 0, 1

T02 In this picture, what are the leafs of the tree?

[

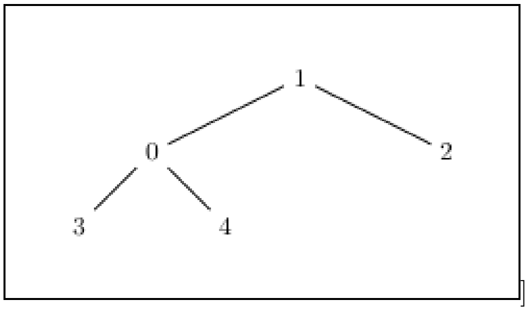


]

1. 2, 3, 4
2. 1
3. 0,1,2, 3, 4
4. 0, 1

T03 We define the `height` H of a binary tree `T` as follows: the height of the empty tree is 0, the tree with 1 node has height 1, and if $T = (r, T_g, T_d), H(T) = 1 + \max(H(T_g), H(T_d))$. What is for the following tree:

[



1. 3
2. 2
3. 1
4. 5

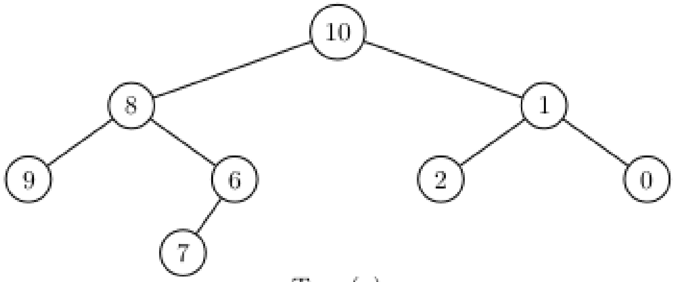
T04 We define the `height` H of a binary tree `T` as follows: the height of the empty tree is 0, the tree with 1 node has height 1, and if $T = (r, T_g, T_d), H(T) = 1 + \max(H(T_g), H(T_d))$. What is the **minimal** value for `H(T)` when `T` has n nodes?

1. $\log_2(n)$
2. n
3. 1
4. $n \log(n)$

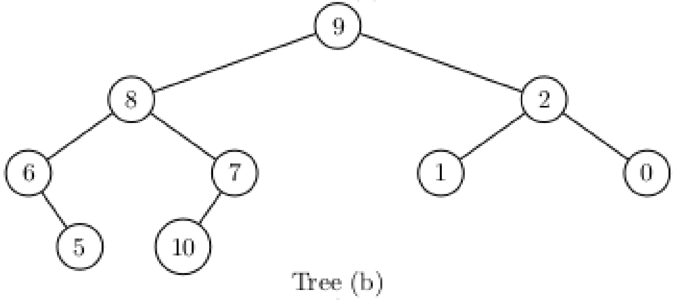
T05 We define the `height` H of a binary tree `T` as follows: the height of the empty tree is 0, the tree with 1 node has height 1, and if $T = (r, T_g, T_d), H(T) = 1 + \max(H(T_g), H(T_d))$. What is the **maximal** value for `H(T)` when `T` has n nodes?

1. $\log_2(n)$
2. n
3. 1
4. $n \log(n)$

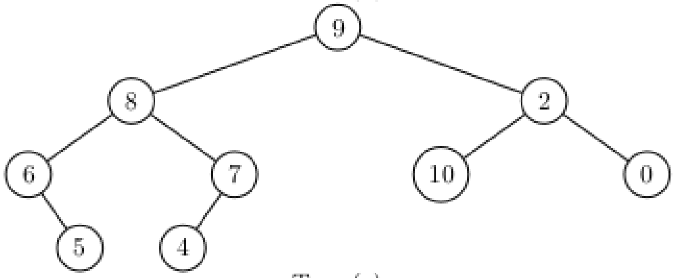
T06 Which of the following trees are binary search trees?



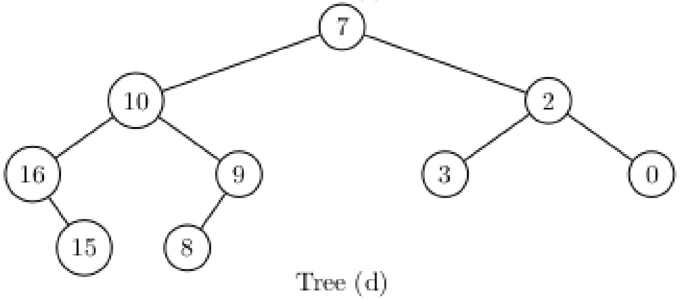
Tree (a)



Tree (b)



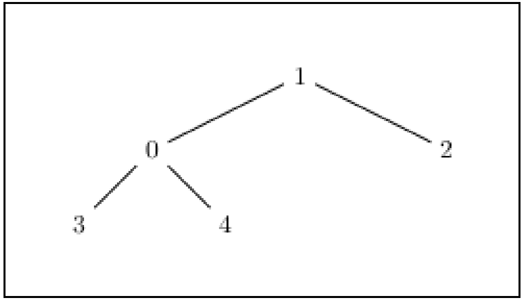
Tree (c)



Tree (d)

1. a, d
2. a, b, c, d
3. a, b
4. a, c, d

T07 Consider the following tree. We suppose we traverse it in prefix order, infix order and postfix order and print the visited nodes at each step. What are the results?



1. 10342; 30412; 34021
2. 01234; 43210;30412
3. 34021;30412;10342
4. 0; 1; 3

L01 Let `L` be a simple linked list with n elements. What is the complexity of adding an element at the head of `L`?

1. $O(1)$
2. $O(n)$
3. $O(\log n)$
4. $O(n^2)$

L02 Let `L` be a simple linked list with n elements. What is the average complexity of searching for an element in `L`?

1. $O(1)$
2. $O(n)$
3. $O(\log n)$
4. $O(n^2)$

L03 Let `L` be a simple linked list with n elements. What is the complexity of inserting an element at the end of `L`?

1. $O(1)$
2. $O(n)$
3. $O(\log n)$
4. $O(n^2)$

L04 Let `L` be a doubly linked list with n elements. What is the complexity of inserting an element at the end of `L`?

1. $O(1)$

2. $O(n)$
3. $O(\log n)$
4. $O(n^2)$

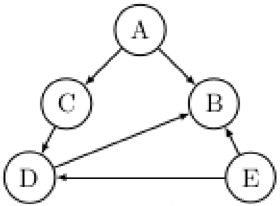
H01 Let `H` be a hash table containing n elements. What is the complexity of inserting a new element?

1. $O(n)$
2. $O(1)$
3. $O(\log n)$
4. $O(n^2)$

H02 Let `H` be a hash table containing n elements. What is the complexity of searching for an element?

1. $O(n)$
2. $O(1)$
3. $O(\log n)$
4. $O(n^2)$

G01 Consider the following graph

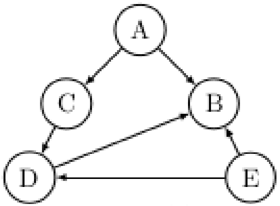


Graph (a)

What are the vertices?

1. A, B, C, D, E
2. A, C, D, E
3. B
4. A, D, E

G02 Consider the following graph

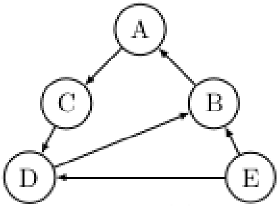


Graph (a)

What are the edges?

1. (A, B); (A, C); (C, D); (D, B); (E, B); (E, D)
2. (B, A); (C, A); (D, C); (B, D); (B, E); (D, E)
3. (A, B); (A, C); (C, D)
4. (D, B); (E, B)

G03 Consider the following graph



Graph (b)

Which of these paths are a circuit?

1. (A, C, D, B, A)
2. (A, C, D, B, A) and (D, B, E, D)
3. (A, C, D, A)
4. (A, B, C, D, E, A)