

CS2040C Semester 1 2018/2019
Data Structures and Algorithms

Tutorial 05 - Trees and Binary Heap

For Week 7 (Week Starting 1 October 2018)

Document is last modified on: September 24, 2018

1 Introduction and Objective

In this tutorial, we will start our discussion on Binary Heap data structure. You may like to review <https://visualgo.net/en/heap> to refresh your memory before the tutorial.

2 Tutorial 05 Questions

Trees

In lecture, you have been introduced to trees. The definition of tree actually results in many interesting and useful properties of trees:

1. All the vertices/nodes are *connected*. This means you can get from one node to another using a sequences of edges, known as a *path*.
2. A tree with N vertices/nodes have **exactly** $N - 1$ edges.
3. A tree is acyclic, which means it does not have a cycle.(Does not form a *ring*).

Q1). Prove or disprove: “There is a unique path between any two distinct nodes of a Tree (a connected acyclic undirected graph)”.

Q2). Prove or disprove: “In a complete binary tree with N nodes ($N > 1$), the number of leaf nodes is strictly less than $N/2$ ”.

Binary Heap

Q3). Give an algorithm to find all vertices bigger than some value x in a max heap that runs in $O(k)$ time where k is the number of vertices in the output.

This is a new algorithm analysis type for most of you as the time complexity of the algorithm does not depend on the input size n but rather the output size k :O...

Note that this question has also been integrated in VisuAlgo Online Quiz, so it may appear in future Online Quizzes :).

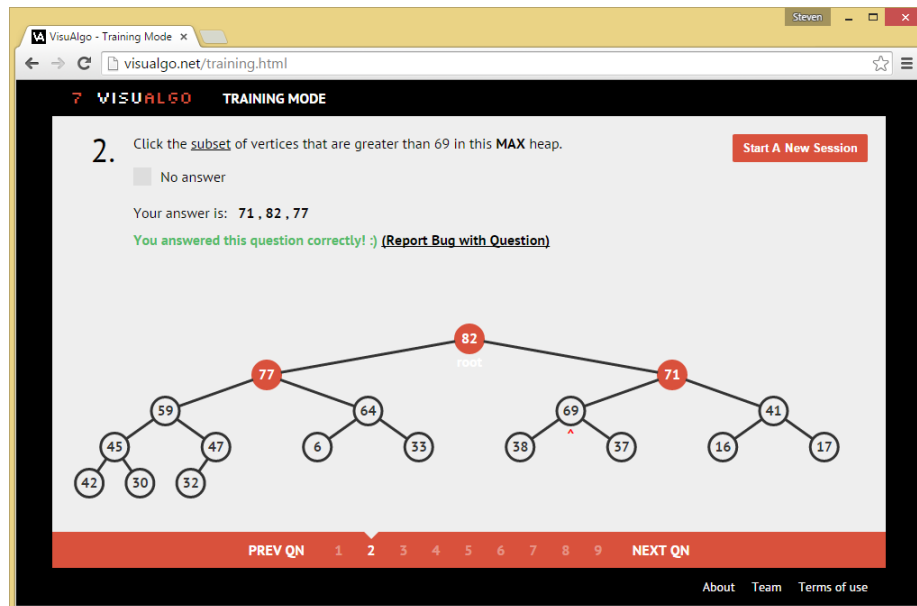


Figure 1: Also automated :)

Q4). The *second* largest element in a max heap with more than two elements (to simplify this question, you can assume that all elements are unique) is always one of the children of the root. Is this true? If yes, show a simple proof. Otherwise, show a counter example.

Problem Set 3

Q5). We will end the tutorial with (early) discussion of (seemingly hard but actually easy) PS3. This single PS3 contains two interesting features of Binary Heap data structure that are not available in C++ STL `priority_queue` and Java `PriorityQueue` yet: `UpdateKey(old_v, new_v)` and `DeleteKey(v)` where v is not necessarily the max element. You may assume that you are provided with pointer/iterator to the node that currently represents key v .

Q6). Refer to Q5. What if the pointer/iterator to the node that currently represents key v is not provided? It is possible to obtain an iterator/pointer to v in the Binary Heap quickly? How?