CS2040C Semester 1 2019/2021

Data Structures and Algorithms

Tutorial 04 - PQ ADT

For Week 06

Document is last modified on: September 9, 2020

1 Introduction and Objective

This tutorial marks the end of the first $\frac{1}{3}$ of CS2040/C/S: Basic Java/C++, basic analysis of algorithms (worst case time complexity only), various sorting algorithms, and various linear Data Structures (DSes): Linked List/Stack/Queue/Doubly Linked List/Deque.

This tutorial marks the start of the next $\frac{1}{3}$ of CS2040/C: Various non-linear DSes. Today, we will discuss the Priority Queue (PQ) ADT with its Binary Heap implementation (use `https://visualgo.net/en/heap` to help you answer some questions in this tutorial).

2 Tutorial 04 Questions

Basic Binary Heap

Q1). Quick check: Let's review all 5 basic operations of Binary Heap (use the Exploration mode of `http://visualgo.net/en/heap`). During the tutorial session, the tutor will randomize the Binary Heap structure, ask student to `Insert(random-integer)`, perform `ExtractMax()` operations (or the first few steps of `HeapSort()`), and/or the $O(N \log N)$ or the $O(N)$ `Create(from-a-random-array)`.

Q2). What is the minimum and maximum number of comparisons between Binary Heap elements required to construct a Binary (Max) Heap of arbitrary n elements using the $O(n)$ `Create(array)`? Note that this question has been integrated in VisuAlgo Online Quiz, so it may appear in future Online Quizzes :).

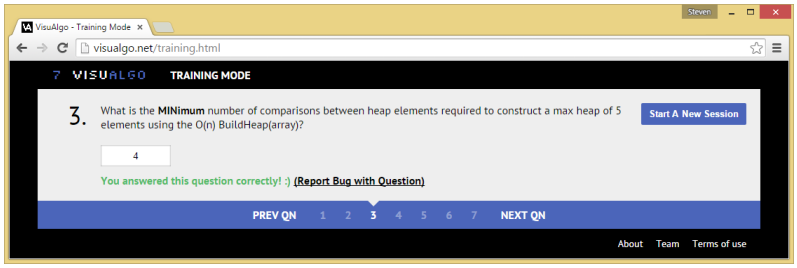

Figure 1: Now automated :)

More Binary Heap

Q3). 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.
Note that this kind of (simple) proof may appear in future CS2040/C written tests, so please refresh your CS1231 (if you have taken that module) or just concentrate on how the tutor will answer this kind of question.

Q4). Give an algorithm to find all vertices that have value $> x$ in a max heap of size n.
Your algorithm must run in $O(k)$ time where k is the number of vertices in the output.
Key lesson: This is a new algorithm analysis type for most of you as the time complexity of the algorithm does not depends 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 :).

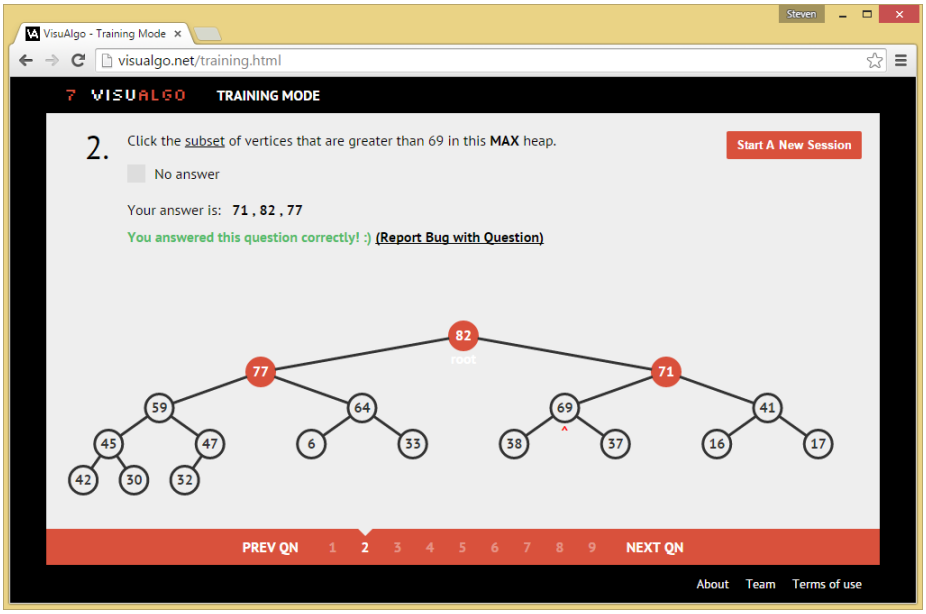

Figure 2: Also automated :)

More ADT PQ Operations

Q5). There are two Binary Heap data structure features that are not available (yet) in C++ STL `priority_queue`, Java `PriorityQueue`, and Python `heapq`: `Increase/Decrease/UpdateKey(old_v, new_v)` and `DeleteKey(v)` where `v` is not necessarily the max element. These two operations are not yet included in VisuAlgo (the hidden slide `https://visualgo.net/en/heap?slide=3-1`).

Q5a). Given `https://www.comp.nus.edu.sg/~stevenha/cs2040c/demos/BinaryHeapDemo.cpp` (that is a Binary Max Heap), what should we modify/add so that we can implement `DecreaseKey(old_v, new_lower_v)`?

Q5b). Given `https://www.comp.nus.edu.sg/~stevenha/cs2040c/demos/BinaryHeapDemo.cpp` (that is a Binary Max Heap), what should we modify/add so that we can implement `DeleteKey(v)` where `v` is not necessarily the max element?

Hands-on 4

TA will run the second half of this session with a few to do list:

- Very quick review of C++ STL std::priority_queue; plus a review of max to min PQ conversion,
- Then, live solve another chosen Kattis problem involving priority queue ADT.

Problem Set 2

We will end the tutorial with **short algorithmic** discussion of PS2 that will due soon.