CS 261 Recitation 7: Binary Heap & Heapsort

In order to get credit for the recitation, you need to be checked off by the end of recitation. For non-zero recitations, you can earn a maximum of 3 points for recitation work completed outside of recitation time, but you must finish this recitation before the next recitation. For extenuating circumstance, contact your recitation TAs and Instructor.

Group work, and individual work are highlighted

Recitation 7 Grade Breakdown:

• Part 1: Array-Based Binary Heap Operations	4 pts
• Part 2: Find kth largest element in an Array	6 pts

Part 1: Array-Based Binary Heap Operations

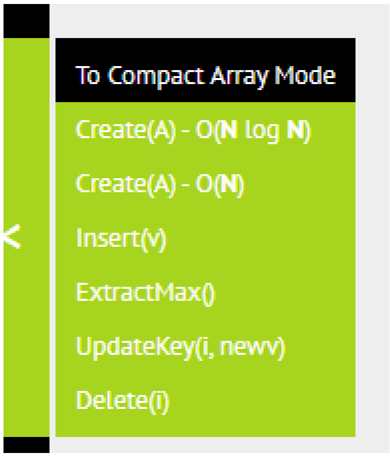
Step 0:

Take a minute to familiarize yourself with the way the array-based Heap works:

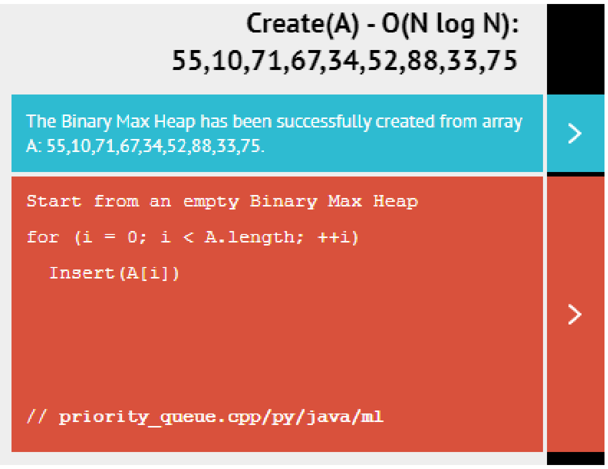
https://visualgo.net/en/heap?slide=1

Note: unlike in our lecture, the binary heap you visualize on this website is a max heap.

You can convert the binary heap to an array by clicking the "To Compact Array Mode" on the bottom left



As the operation runs, you will see the site step through a pseudocode implementation of the operation in the box at the lower right of the page:



Create(A) - O(N log N):
55,10,71,67,34,52,88,33,75

The Binary Max Heap has been successfully created from array A: 55,10,71,67,34,52,88,33,75.

```
Start from an empty Binary Max Heap
for (i = 0; i < A.length; ++i)
  Insert(A[i])



// priority_queue.cpp/py/java/ml
```

Play around a bit with the operations of Binary Heap so you understand how the site works.

(2 pts) Step 1:

Create a new, empty binary heap, and run the following sequence of operations.

```
insert(55)

insert(10)

insert(71)

insert(67)

insert(34)

insert(52)

insert(88)

insert(33)

insert(75)
```

Make note of how the heap changes as it's being built and what it looks like when all the operations are complete.

Take a screenshot of the final Binary Heap.

(2 pts) Step 2:

Convert the Binary Heap that you built in Step 1 into an array, and apply a heapsort in ascending order. (Make sure you understand how heapsort works)

Take a screenshot of the array before and after sorting.

(6 pts) Part 2: Find kth largest element in an Array

Download and unzip the start code: (wget command is recommended)

wget https://classes.engr.oregonstate.edu/eecs/summer2022/cs261-001/recitations/rec7.zip

Problem statement: Given an integer array `nums` and an integer `k`, **use heapsort** to return the `k`th largest element in the array.

Note that it is the `k`th largest element in the sorted order, not the `k`th distinct element.

In `kth_largest.c`, modify the program so that the `findKthLargest()` returns the `k`th largest element in the integer array `nums`.

Feel free to add any helper functions if needed.

Constraints:

- 1 <= `k` <= `nums.length` <= 10^4
- -10^4 <= `nums[i]` <= 10^4
- You are NOT allowed to use any built-in sorting functions (i.e., `sort()`)

Make sure you get checked off by showing them the output of your program, your report, and your group work before the end of your recitation section.

For backup purposes, please submit your work for this recitation (including all documents/text files for group work, and programs) to TEACH.