Data Structures & Algorithms

Heaps-II

B.Bhuvaneswaran, AP (SG) / CSE

9791519152

bhuvaneswaran@rajalakshmi.edu.in

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Top k

- One common type of interview problem is one that asks you to find the k best elements, with "best" being defined by the problem.
- The easiest way to solve these problems is to just sort the input according to the criteria defined in the problem, and then return the top k elements.
- This has a time complexity of O(n logn) if n is the length of the input.

Top k

- Using a heap, we can instead find the top k elements in O(n logk).
- Logically, k < n, so this is an improvement. Practically, because log is so fast anyway, it probably isn't a big deal in terms of a speed increase.
- But when interviewers give you these kinds of problems, it is these small improvements that they are looking for.

What is the improvement?

- Create a max heap at the start, iterate over the input while pushing every element on the heap (according to the problem's criteria), and pop from the heap once the size exceeds k.
- Because the heap's size is bounded by k, then all heap operations are at worst O(logk).
- Multiply this by n iterations to get O(n logk).
- Because we are using a max heap and we are popping from the heap according to the problem criteria, pops remove the "worst" elements, so at the end, the k "best" elements will remain in the heap.

Top K Frequent Elements

- Given an integer array nums and an integer k, return the k most frequent elements.
- It is guaranteed that the answer is unique.

Example

- Input:
  - nums = [1, 1, 1, 2, 2, 3]
  - k = 2
- Output:
  - [2, 1]

Find K Closest Elements

- Given a sorted integer array arr, two integers k and x, return the k closest integers to x.
- The answer should also be sorted in ascending order.
- If there are ties, take the smaller elements.

Example

- Input:
  - nums = [1, 4, 10, 15, 22]
  - k = 3
  - x = 11
- Output:
  - [4, 10, 15]

Queries?

Thank You…!