Congratulations! You passed!
TO PASS 1% or higher

Keep Learning

GRADE
100%

Interview Questions: Priority Queues (ungraded)

TOTAL POINTS 3

1. **Dynamic median.** Design a data type that supports *insert* in logarithmic time, *find-the-median* in constant time, and *remove-the-median* in logarithmic time. If the number of keys in the data type is even, find/remove the *lower median*. **1 / 1 point**

   Note: these interview questions are ungraded and purely for your own enrichment. To get a hint, submit a solution.

   Dynamic median
   Specification:
   1. insert in logarithmic time
   2. find-the-median in constant time
   3. remove-the-median in logarithmic time

   Correct
   Hint: maintain *two* binary heaps, one that is max-oriented and one that is min-oriented.

2. **Randomized priority queue.** Describe how to add the methods `sample()` and `delRandom()` to our binary heap implementation. The two methods return a key that is chosen uniformly at random among the remaining keys, with the latter method also removing that key. The `sample()` method should take constant time; the `delRandom()` method should take logarithmic time. Do not worry about resizing the underlying array. **1 / 1 point**

   Randomized priority queue:
   Specification:
   Add two methods to binary heap implementation:
   1. sample() method, choose uniformly at random among the remaining keys. (constant time)
   2. delRandom() also removing that key. (logarithmic time)

   Correct
   Hint: use `sink()` and `swim()`.

3. **Taxicab numbers.** A *taxicab* number is an integer that can be expressed as the sum of two cubes of positive integers in two different ways: $a^3 + b^3 = c^3 + d^3$. For example, 1729 is the smallest taxicab number: $9^3 + 10^3 = 1^3 + 12^3$. Design an algorithm to find all taxicab numbers with a, b, c, and d less than n. **1 / 1 point**

   - Version 1: Use time proportional to $n^2 \log n$ and space proportional to n^2.
   - Version 2: Use time proportional to $n^2 \log n$ and space proportional to n.

   Taxicab numbers:
   Design an algorithm to find all taxicab numbers with a, b, c, and d less than n.

   V1: Use time proportional to n^2logn and space proportional to n^2.
   V2: Use time proportional to n^2logn and space proportional to n.

   Correct
   Hints:

   - Version 1: Form the sums $a^3 + b^3$ and sort.
   - Version 2: Use a min-oriented priority queue with n items.