**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*.

**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.

**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, 17291729 is the smallest taxicab number: 9^3 + 10^3 = 1^3 + 12^393+103=13+123. Design an algorithm to find all taxicab numbers with *a*, *b*, *c*, and *d* less than *n*.

* Version 1: Use time proportional to *n*2log*n* and space proportional to *n*2.
* Version 2: Use time proportional to *n*2log*n* and space proportional to *n*.