Indian Institute of Technology Kharagpur

CS29003: Algorithms Laboratory, Spring 2021

Assignment 7: Heaps and Priority Queues

2PM – 5PM 23rd March, 2021

General Instructions (to be followed strictly)

Submit a single C/C++ source file.
Do not use global variables unless you are explicitly instructed so.
Do not use Standard Template Library (STL) of C++.
Use proper indentation in your code and comment.
Name your file as `<roll_no>_<assignment_no>`.
Write your name, roll number, and assignment number at the beginning of your program.

Numbers (positive integers, to be precise) that can be expressed as sum of two cubes of positive integers in at least k different ways are called *k-way Ramanujan numbers*. For example, the number 1729 is a 2-way Ramanujan number (the smallest) which can be expressed as $1^3 + 12^3$ and $9^3 + 10^3$. The smallest 3-way Ramanujan number is $87539319 = 228^3 + 423^3 = 167^3 + 436^3 = 255^3 + 414^3$. In this assignment, we are interested in finding all 2-way Ramanujan numbers (also called *taxicab numbers*) in a specific range.

Define a min-heap consisting of nodes of triples $(a, b, a^3 + b^3)$ with the third component as the key. Implement the following functions for the heap:

- *build-heap*: returns a new empty heap
- *remove-min*: removes (and returns) the item with smallest key and restores the heap property
- *insert*: inserts a new triple maintaining the heap property

Write a function that, on input a positive integer n, finds all possible sums of cubes $a^3 + b^3$ (where $a, b \in [0, n]$) in sorted order using no more than $O(n)$ space.[1] Use the minimum priority queue defined above. Initially populate the heap with triples $(i, 0, i^3)$ for $i = 0, 1, 2, \ldots, n$. Until the queue becomes empty, do the following: remove the triple $(a, b, a^3 + b^3)$ and then, if $b < n$, insert the item $(a, b + 1, a^3 + (b + 1)^3)$. Every time a triple is removed, perform a check to see if you have found a 2-way Ramanujan number. If so, print the number (say, r) along with the 4 integers a, b, c, d such that $a^3 + b^3 = c^3 + d^3 = r$. You may print in the format $r\ (a, b)\ (c, d)$ in a separate line.

Use the above function to find and print all 2-way Ramanujan numbers in the range $[0, M]$. Here, $M \leq 2^{60}$ is a positive integer entered by the user.

In the *main()* function,

- Read a positive integer M ($\leq 2^{60}$).
- Print all 2-way Ramanujan numbers in the range $[0, M]$.

Do not use any built-in library functions.

[1] The problem can be solved using $O(n^2)$ space by storing all n^2 possible sums in an array and then sorting it.

Sample Output

```
Enter M: 35000

 1729 ( 1,12) ( 9,10)
 4104 ( 2,16) ( 9,15)
13832 ( 2,24) (18,20)
20683 (10,27) (19,24)
32832 ( 4,32) (18,30)
```