Exercise 1 ....

Computational number theory. Write a program that prints out all integers of the form $a^3 + b^3$ where a and b are integers between 0 and n in sorted order, without using excessive space. That is, instead of computing an array of the n^2 sums and sorting them, build a minimum-oriented priority queue, initially containing $(0^3, 0, 0)$, $(1^3 + 1^3, 1, 1)$, $(2^3 + 2^3, 2, 2)$,... , $(n^3 + n^3, n, n)$. Then, while the priority queue is nonempty, remove the smallest item $(i^3 + j^3, i, j)$, print it, and then, if $j < n$, insert the item $(i^3 + (j + 1)^3, i, j + 1)$. Use this program to find all distinct integers a, b, c, and d between 0 and 10^6 such that $a^3 + b^3 = c^3 + d^3$, such as $1729 = 93 + 103 = 13 + 123$.

Exercise 2 ....

Dynamic-median finding. Design a data type that supports insert in logarithmic time, find the median in constant time, and remove the median in logarithmic time.

Hint: Keep the median key in v; use a max-oriented heap for keys less than the key of v; use a min-oriented heap for keys greater than the key of v. To insert, add the new key into the appropriate heap, replace v with the key extracted from that heap.