Gems from the world of data structures and algorithms

Binary search using only addition and subtraction only

Difficulty level: *high*

???

Design an algorithm that, given an array of n distinct numbers in ascending order, determines whether a given number is in the array. The only arithmetic operations allowed are additions and subtractions. The running time of your algorithm should be $O(\log n)$ in the worst case.
(Avoid reading the hints given below unless you have tried really hard.)

Hint: Try to get a better insight into the Binary Search algorithm which you know. Asking right kind of questions will help here again.

- How crucial is the *division* operation in the usual Binary Search algorithm ?

  First you may design an algorithm which runs in $O(\log n)$ time but uses $O(\log n)$ extra space only. This is the first solution on the way to our final solution of the problem. You may transform this solution into an $O(\log^2 n)$ time algorithm with $O(1)$ space.

- How crucial is that we always go to the middle element of the current search interval during the usual Binary Search algorithm?

  It will still be fine if we go to an *almost middle* of the current search interval. The problem we face in getting rid of the $O(\log n)$ space in the first solution is that we do not have an efficient mechanism to compute all numbers which we store in $O(\log n)$ space. Can you use some well known numbers (*********) here ?

How To Make A Chrysanthemum Mirror
(out of plastic spoons)
by Addicted2Decorating.com

every art is beautiful and so is the art of algorithm design ...