A Challenging Math Problem: *n* Gas Stations on a Circular Road

Problem: On a circular road there are n gas stations, located at arbitrary points (not equidistant). The total amount of gas in the gas stations is just enough to go once around the circle. You have a car with no gas. Prove that there exists a gas station which you can start at, and successfully go all the way around the circle in the clockwise direction.
Note: A version of this problem, in which an algorithm for finding the starting gas station is sought, constitutes a popular interview question in computing, the interviewer's goal being to see if the applicant can come up with anything better than the obvious $O(n^2)$-time algorithm. In this math puzzle, only the existence of a starting point, and not its identity, is in question.

Solution: It is best to characterize each gas station by a signed number, corresponding to the amount of excess gas available there compared with how much is needed to get to the next station. For example, at the outset the distribution of gas among 6 stations may be:

A:+0.5 B:–0.3 C: –0.7 D:+1.1 E: –0.4 F: –0.2
(The measurement unit employed does not matter.)

Imagine that the first number (for station A) follows the last one (for station F), that is, we have a circular list. Given that the total amount of gas is just enough for one trip around, these numbers must sum to 0. Now, any station with a deficit (negative number) cannot be the starting point. We can eliminate all such stations by combining their numbers with a previous positive number, where "previous" means in the circular counterclockwise order. So, the example above is simplified to:

A: –0.5 D:+0.5
(e.g., the deficits at B and C are combined with the excess gas at A, giving it a deficit of 0.5)

In one more step, A is eliminated, and we get D:+0.0, indicating that D is the only possible starting station. Such a station must exist, because the sum of the positive numbers has the same magnitude as the sum of the negative numbers. The process just described can provide an $O(n)$-time algorithm for identifying a possible starting station (there may be more than one possible starting point). Note that direct implementation of the process described above leads to $O(n^2)$ complexity, because there may be just a single negative entry, whose assimilation requires $n - 1$ phases.

Algorithm: Let n be a power of 2; if not, we can pad the list with 0-supply gas stations which do not affect the solution. The key idea is to try to combine information from two consecutive points into a single point. If this can be done in n steps in the initial state of n candidate gas stations, the total complexity will be $n + n/2 + n/4 + \ldots = 2n - 1 = O(n)$. In what follows, all variables represent nonnegative numbers, so $+x$ is nonnegative and $-y$ is nonpositive. There are four cases for combining the info for two gas stations: (1) A:$+x$ and B:$+y$ are combined into A:$+(x + y)$. Note that if B can be a starting point, then so can A, so no information is lost by only carrying A forward to the next step. (2) A:$-x$ and B:$-y$ are combined into A:$-(x + y)$. Neither A nor B can be a starting point, so the total deficit is carried forward with A. (3) A:x and B:$-y$ are combined into A:$(x - y)$. This is obvious, based on the existence argument presented earlier. The fourth case is more tricky. (4) A: $-x$ and B:$+y$ are combined into $[-x]$B:y. The notation here is that B is the only viable candidate as a starting point, so it is carried forward, but we also remember that there was a deficit of x just before B. We now have to modify the combining rules above to take deficits of the preceding kind into account. This is straightforward, so I just give one example case. (1') $[-u]$A:$+x$ and $[-v]$B:$+y$ are combined differently according to the following conditions. If $x \geq v$, then the combined point becomes $[-u]$A:$(x - v + y)$. If $x < v$, then A cannot be a starting point and we combine into $[-u + x - v]$B:$+y$. To summarize, we combine a pair of consecutive points into a single candidate point, with a preceding deficit and a following excess.

Example: Let me clarify the workings of my $O(n)$-time algorithm above by means of an example. Let's take the six-station example that I had presented earlier and pad the list with two stations G and H with 0 excess gas, to make the number of points a power of 2.

A:+0.5 B:–0.3 C: –0.7 D:+1.1 E: –0.4 F: –0.2 G:+0 H:+0

I don't know if G or H can ever end up being the final answer, but if they do, the correct answer is the next real gas station, A. Pairwise combining, A with B, C with D, E with F, and G with H yields:

[–0]A:+0.2 [–0.7]D:+1.1 [–0]E:–0.6 [–0]G:+0

Note that A, D, E, and G are the remaining candidate gas stations for the starting point. The [–0] preceding a station name means that the combined stations do not have a deficit before the named station, whereas [–0.7] before D means that by using D as a representative of the C-D group, we are carrying a deficit of 0.7 before D. The next combining step, A with D and E with G, yields:

[–0.5]D:+1.1 [–0.6]G:+0

And the final combining step produces:

[–0.5]D:+0.5

Thus, we have identified D as a possible starting gas station.