Hit the Number

It's time to show you how the Codility challenge code-named Rho can be solved. You can still give it a try, but no certificate will be granted. In this task we are looking for the shortest sequence $x_1, \ldots, x_n$, such that:

- $x_0 = 1$,
- $x_n = A$, and
- for each $i = 1, \ldots, n$, $x_i = x_j + x_k$ for some $0 \leqslant j, k < i$.

There are three basic programming techniques that can be applied in tasks of this type. From the fastest and most limited to the slowest and most general, they are:

- greedy programming,
- dynamic programming, and
- back-tracking.

But which should be applied here?

Greedy solution

Let us start with a greedy approach. We can easily construct a sequence of logarithmic length satisfying the above conditions, starting from its end. We start with $x_n = A$. If x_i is even, then $x_{i-1} = \frac{x_i}{2}$. Otherwise $x_{i-1} = x_i - 1 = x_i - x_1$. For example, for $A = 15$ we obtain $[1, 2, 3, 6, 7, 14, 15]$.

Another greedy approach, producing sequences of the same length, is as follows. The sequence starts with powers of 2: $x_0 = 1, x_1 = 2, x_2 = 4, \ldots, x_k = 2^k$, where $2^{k+1} > A$ and $2^k \leqslant A$. Let us consider a binary representation of A. There are at most $k + 1$ 1s. So, in at most k steps, we can sum up corresponding powers of 2 and obtain a complete sequence. For example, for $A = 15$ we obtain $[1, 2, 4, 8, 12, 14, 15]$.

Both of the above approaches fail to produce the shortest possible sequences; for $A = 15$ we can take the sequence $[1, 2, 3, 6, 12, 15]$. So, one should refrain from being greedy because the greedy algorithm is incorrect. However, the approaches provide an upper bound on the length of the shortest possible sequence: $n \leqslant 2 \cdot \lfloor \log_2 A \rfloor$. On the other hand, such sequences cannot grow faster than powers of 2, and $x_i \leqslant 2^i$; hence $n \geqslant \lceil \log_2 A \rceil$.

Dynamic solution

How about the dynamic approach? A natural parameterization of the problem is to calculate the shortest sequences ending with a given integer B, for all $B = 1, 2, \ldots, A$. For $B = 1$, such a sequence comprises just one number: 1. For larger values of B we can use the sequences computed for smaller values of B, and look for a sequence that contains both some number x and $B - x$. The shortest such sequence can be extended by B and memorized. Such an algorithm works in $O(A^2 \cdot \log A)$ time. Unfortunately, it is also incorrect. For example, for $A = 77$ it produces the sequence $[1, 2, 3, 6, 9, 11, 22, 44, 55, 77]$, whereas the shortest possible sequence is $[1, 2, 4, 5, 9, 18, 36, 41, 77]$.

Back-tracking solution

The only approach that remains is back-tracking. This checks all possible sequences, and hence is bound to be correct. Unfortunately, its running time is exponential (even in terms of A). The actual cost depends on the number of sequences considered.

How can we limit the number of sequences to consider? One natural observation is that we can limit ourselves to increasing sequences. Obviously, if any number appears in the sequence more than once, it cannot be optimal. Moreover, any non-monotonic sequence, containing different integers, can be rearranged in such a way that it becomes increasing.

Another optimization is to limit the length of considered sequences. We know that the length does not exceed $n \leqslant 2 \cdot \lfloor \log_2 A \rfloor$, but for some numbers it can be smaller. In particular, for powers of 2 we have $n = \log_2 A$. So, we can start with a sequence built by one of the greedy algorithms presented above, and look only for shorter sequences.

It is tempting to assume that each element of the sequence can be constructed as a sum of the preceding element and some other element. This significantly reduces the number of sequences to consider. On the one hand, such an assumption is incorrect; for example, the shortest sequence for $A = 12\,509$ is:

$$[1, 2, 3, 6, 12, 13, 24, 48, 96, 192, 384, 397, 781, 1\,562, 3\,124, 6\,248, 6\,261, 12\,509].$$

The number 13 is needed to obtain 397 and $6\,261$, but it is followed by 24, which is obtained from 12 alone.

However, on the other hand, it is the smallest counterexample, and it is far beyond the maximum value of A. So, this optimization is also acceptable.