Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department of Computer Science
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger
Tomáš Gavenčiak

27th October 2016

Data Structures & Algorithm Solutions to Sheet P5 AS 16

Solution P5.1 *Binary function value search.*

The solution was to use binary search for x in the range $0 \dots x_{max}$, comparing a to $f(x)$ instead to `data[x]` as you would in a sorted array binary search. Otherwise the problem was essentially the same.

On the lecture website, you can find both non-recursive (`while`-loop based) and recursive solution, both running in time $\mathcal{O}(n \log x_{max})$ and both implementing the algorithms from the lecture notes. Their source contains further comments on the implementation.

The time limit was set such that just computing all the values of $f(x)$ for $x = 0 \dots x_{max}$ even just once would take too long, so a binary search is necessary. Observe that since $\log_2(x_{max}) \simeq 24.25$, less than 30 calls to f are necessary for any input, and contrast that to $x_{max} = 2 \cdot 10^7$ calls required to a linear scan of the range.

Data

`judge1` $n = 11$ values including 0 and 1 (the lowest possible values), $500000000 = f(x_{max})$ and $500000001 = f(x_{max}) + 1$ and some random values in between.

Notes on submitted solutions. The most common problem was the input value $f(x_{max})$. Some solutions maintained a search range with variables such as `min` and `max`, but only worked for values `min .. max - 1`. While it can be natural to work with these conditions, the solutions also set the starting search range to `min = 0` and `max = x_max`, missing the maximal value.

A second problem was that some solutions contained a loop like `while (min != max) {...}`, but then they would also somehow set `max = min - 1` during the search, e.g. via `max = (max + min) / 2 - 1` after having `max == min + 1`. This would result in an infinite loop. One valid solution is to replace `(min != max)` with `(min < max)`. Note that infinite recursion fails with `RUNTIME ERROR` in the judge.

Notes on the function f. (Extra material, not necessary to solve the problem)

The function is actually $f(x) = x\,l(x)$, where $l(x) = \lfloor \log_2 x \rfloor + 1$, which is the number of binary digits of x. Knowing this was in no way important to solve the problem, but it is theoretically possible to compute the inverse of f almost directly. (Again, you are encouraged to try.) However, it is certainly not easier than the binary search program and would be significantly faster only for very large x_{max}.