Elements of Programming Interviews
Task 16.7
Proof

Oleksii Symon

January 07, 2018

Task

A string is said to be palindromic if it reads the same backwards and forwards. A decomposition of a string is a set of strings whose concatenation is the string. Suppose that you are given a string and you have to find the worst-case number of operations required to find all palindromic decompositions of this string.

Solution

To find all palindromic decomposition of a string using recursion, we use prefixes of length $1, 2 \ldots n$ for the first string in decomposition and then recursively compute the decomposition of the corresponding suffix. This can be written as a recurrence relation.

$$g(n) = 1 + \sum_{i=1}^{n-1} g(i)$$

Then we can write out some expression values and look for dependencies

$$\begin{aligned}
&g(1) = 1 \\
&g(2) = 1 + g(1) = 2 \\
&g(3) = 1 + g(2) + g(1) = 4 \\
&g(4) = 1 + g(3) + g(2) + g(1) = 8
\end{aligned}$$

At this point we can see a pattern. Lets say that $g(n) = 2^{n-1}$ and try to prove this by induction

$$
\begin{aligned}
g(n) &= 1 + \sum_{i=1}^{n-1} g(i) = 2^{n-1} \\
g(n+1) &= 1 + \sum_{i=1}^{n} g(i) \\
&= \mathbf{1} + \left(\sum_{\mathbf{i=1}}^{\mathbf{n-1}} \mathbf{g(i)} \right) + g(n) \\
&= g(n) + g(n) = 2g(n) \\
&= 2 * 2^{n-1} = 2^n
\end{aligned}
$$

So the worst-case number of operations required is for string with all equal symbols "*aaaaa*" and equals $O(2^{n-1})$.