Problem Tutorial: "Little LCS"

First let's note that LCS is always at least n. Indeed, for every i, there is some character that's present in both (s_{2i-1}, s_{2i}) and (t_{2i-1}, t_{2i}). It hints that awesome strings must have some very special structure.

And indeed, the problem is mainly about understanding the structure of awesome strings, the rest is just implementing straightforward checking.

It turns out that there are 2 classes of pairs of awesome strings:

- One string has form `ABABAB...ABA`, and the second one has `C` at all odd positions and at every even position has one of `A`, `B`. (And, obviously, same thing over all other permutations of `A`, `B`, `C`).

  It's obvious that strings of such format have LCS of length n: it can't be higher as LCS can't contain any of $n+1$ `A`s of s.

- Both strings have `C` at all even positions, and for some k, first k odd positions of s contain `A`, last $n+1-k$ odd positions of s contain `B`, first k odd positions of t contain `B`, last $n+1-k$ odd positions of t contain `A`. (And, obviously, same thing over all other permutations of `A`, `B`, `C`).

  Here it's a bit harder to see why these strings have LCS n, but it can be proved by induction. Indeed, suppose that both s and t contain some string lcs of length $n+1$ as a subsequence. If $lcs[1] =$`C`, then $s[3:2n+1]$ and $t[3:2n+1]$ both have to contain $lcs[2:n+1]$, but their LCS is $n-1$ by induction assumption, so $lcs[1] \neq$`C`. Similarly, $lcs[n+1] \neq$`C`.

  Wlog $lcs[1] =$`A`. Then $lcs[n+1]$ can't be `B`, as t doesn't contain `AB` as a subsequence, so $lcs[n+1] =$ `A`. But this means that $|lcs| \le min(2k-1, 2(n+1-k)-1)$. As $(2k-1)+(2(n+1-k)-1) = 2n$, we get $|lcs| \le n$.

During the contest, you can observe this pattern by just bruteforcing amazing strings for $n \le 4$, and don't have to prove it, but for the completeness of the editorial we will provide the proof here.

So, suppose that (s, t) is a pair of amazing strings of length $2n+1$. Let's show that they are of one of the types above, by induction by n. Base for $n = 1$ can be checked by hand, now we suppose that $n \ge 2$ and the statement is proved for all $n_1 < n$.

Note that $(s[1:2n-1], t[1:2n-1])$ is an amazing pair of strings for $n-1$. Therefore, they are one of the types above. Same goes for $(s[3:2n+1], t[3:2n+1])$.

Suppose that $(s[1:2n-1], t[1:2n-1])$ is an amazing pair of the first type. Wlog $s[1:2n-1] =$ `ABAB...ABA`, and $[1:2n-1]$ has `C` at all odd positions. $s[2n-1:2n+1]$ and $t[2n-1:2n+1]$ have to have LCS 1, and there are only 4 such pairs of strings: (`ABA`, `CBC`), (`ABA`, `CAC`), (`ABC`, `CBA`), (`ACA`, `CBC`). Note that the first two pairs correspond to the pattern of amazing strings of type 1. Let's look at last two cases.

Suppose that $(s[2n-1:2n+1], t[2n-1:2n+1]) =$ (`ABC`, `CBA`). Then $(s[3:2n+1], t[3:2n+1])$ can't be an amazing pair of type 1, so it's an amazing pair of type 2, and all letters at even positions at them are `B`. So, we get $s =$`ABAB...ABABC` and $t =$`C?CBC...CBCBA`. If `?` is replaced with `B`, we get an amazing pair of second type, otherwise it's replaced with `A` and there is a common subsequence of length $n+1$ `ABB...BBA`.

Now suppose that $(s[2n-1:2n+1], t[2n-1:2n+1]) =$ (`ACA`, `CBC`). Then $(s[3:2n+1], t[3:2n+1])$ can't be an amazing pair of type 2, so it's an amazing pair of type 1. As not all letters at even positions at $s[3:2n+1]$ are the same, all letters at even positions of $t[3:2n+1]$ must be the same, so $t[3:2n+1] =$ `CBCB...CBC`. Therefore, we get $s =$`ABAB...ABACA` and $t =$`C?CBC...CBCBC`. If `?` is replaced with `B`, we get an amazing pair of the first type, otherwise it's replaced with `A` and there is a common subsequence of length $n+1$ `ABB...BBC`.

In this case, we proved the statement. Now suppose that $(s[1:2n-1], t[1:2n-1])$ is an amazing pair of the second type. Similarly, $(s[3:2n+1], t[3:2n+1])$ is an amazing pair of the second type. Then, there are two cases: or (s, t) is also an amazing pair of the second type, or the strings have form (`ACBCB...CBCBCA`, `BCACA...CACACB`). But in this case they both contain a string `ACCC...CCB` of length $n+1$ as a subsequence.

We checked all the cases, so congrats to us.