Code Jam 2022 - Round 1A

Analysis: Double or One Thing

For any string $\mathbf{S}$, the number of new strings we can obtain from it is at most $2^{|\mathbf{S}|}$ because for each character in $\mathbf{S}$, there are 2 choices: to highlight it or not.

Test Set 1

Since the length of $\mathbf{S}$ is at most 10, the number of new strings we can obtain is at most 2^{10}. We can enumerate all of them to find the lexicographically smallest one. The time complexity of this solution is $O(2^{|\mathbf{S}|} \times |\mathbf{S}|)$ because there are $2^{|\mathbf{S}|}$ strings in total to compare, and the length of them is at most $2 \times |\mathbf{S}|$.

Test Set 2

Now that the length of $\mathbf{S}$ can be up to 100, 2^{100} is too large to enumerate all of them.

Note that a string p appears before a different string q in alphabetical order if p is a prefix of q or if p has a letter lexicographically smaller at the leftmost position at which p and q differ.

Then, for any character $\mathbf{S_i}$ in $\mathbf{S}$, we have the following rule to decide whether to highlight it or not:

- If the next different character in $\mathbf{S}$ is lexicographically larger than the current character (in other words, there is an index j that $\mathbf{S_j} > \mathbf{S_i}$ and $\mathbf{S_k} = \mathbf{S_i}$ for all $i < k < j$), then we must highlight $\mathbf{S_i}$. That is because when we double $\mathbf{S_i}$, we also push $\mathbf{S_j}$ right and replace that index with $\mathbf{S_i}$ at the same time. Therefore doubling $\mathbf{S_i}$ will make the new string lexicographically smaller.
- If the next different character in $\mathbf{S}$ is lexicographically smaller than the current character (in other words, there is an index j that $\mathbf{S_j} < \mathbf{S_i}$ and $\mathbf{S_k} = \mathbf{S_i}$ for all $i < k < j$), then we must NOT highlight $\mathbf{S_i}$. That is because when we double $\mathbf{S_i}$, we also push $\mathbf{S_j}$ right and replace that index with $\mathbf{S_i}$ at the same time. Therefore doubling $\mathbf{S_i}$ will make the new string lexicographically larger.
- If there is no next characters different from $\mathbf{S_i}$ (in other words, i is the last index in $\mathbf{S}$ or $\mathbf{S_k} = \mathbf{S_i}$ for all $i < k < |\mathbf{S}|$), then we must NOT highlight $\mathbf{S_i}$. That is because doubling $\mathbf{S_i}$ will make the original string be the prefix of the new string, which means the new string is lexicographically larger.

There are a lot of different ways to implement it and here is one: We can preprocess the given $\mathbf{S}$ into groups of same and continuous characters. For example, `BOOKKEEPER` is preprocessed as `[(B,1), (O,2), (K,2), (E,2), (P,1), (E,1), (R,1)]`. Then for each element in this list, we output the character with twice its original occurrence if the character of the next element is lexicographically larger than it. Otherwise, we output the character with its original occurrence.

The time complexity of this solution is $O(|\mathbf{S}|)$ because we need to iterate through $\mathbf{S}$ when preprocessing it, and iterate through the preprocessed list with length up to $|\mathbf{S}|$ when outputing the final answer.