

B: Bewitched Broomstick

Problem author: Yidi Zang

Problem

- Given a string s ($|s| \leq 2 \cdot 10^5$) and a length ℓ .
- Append $\ell - 1$ characters to the string.
- Maximize the appearance of the most frequent substring of length ℓ .

B: Bewitched Broomstick

Problem author: Yidi Zang

Problem

- Given a string s ($|s| \leq 2 \cdot 10^5$) and a length ℓ .
- Append $\ell - 1$ characters to the string.
- Maximize the appearance of the most frequent substring of length ℓ .

Solution Idea

- First try every substring t of length ℓ .
- Find the maximum overlap ℓ of t and a suffix of s .
- After that the remaining characters repeat the period of t .
- Count of the number of appearances and choose the maximum.
- For shorter substring of the suffix they just repeat their period.
- How to find maximum overlap and period of t ?

B: Bewitched Broomstick

Problem author: Yidi Zang

Solution Idea

- First try every substring t of length ℓ .
- Find the maximum overlap ℓ of t and a suffix of s .
- After that the remaining characters repeat the period of t .
- Keep count of the number of appearances and choose the maximum.
- How to find maximum overlap and period of t ?

Maximum Overlap

- Maximum overlap is the “easy” part.
- Reverse the string and add a ‘#’ at the ℓ th position.
- Find the longest match with KMP prefix function.

B: Bewitched Broomstick

Problem author: Yidi Zang

Period of Substrings

- How find the period of a single string?
- Find the first position i where $i + |\text{common prefix}| \geq \ell$.
- Using z-function for each substring is too slow.

B: Bewitched Broomstick

Problem author: Yidi Zang

Period of Substrings

- How find the period of a single string?
- Find the first position i where $i + |\text{common prefix}| \geq \ell$.
- Using z-function for each substring is too slow.
- Instead use Suffix Tree, that is a trie of all suffixes.
- The depth of a node is the length of the common prefix.
- The period can be calculated with a smaller into larger merging of (implicit) segment trees.
- The (annoying) details are left as an exercise to the reader.
- Total Runtime: $\mathcal{O}(n \log^2(n))$.

Notes

- There are also $\mathcal{O}(n\sqrt{n})$ solutions.
- The finding maximum overlap step can also be done in the same Suffix Tree.