

As announced, Java programming is a central part of the project group and arrays, trees, and threads are required skills,

I will give you programming exercises to prepare for the project group, and here is the first one.

Programming task 1:

Text search engines have to support fast search of substrings.

For this purpose, write a Java program

1.1. that reads a string S and computes and stores all rotations $R(i)$ of S in an array R , such that the i -th rotation is stored in $R[i]$.

For example, if $S = \text{„car“}$, $R[1] = \text{„car“}$, $R[2] = \text{„arc“}$, $R[3] = \text{„rca“}$.

1.2. that computes and outputs a sorted array Sort of rotations, i.e.

$\text{Sort}[1] = \text{„arc“}$, $\text{Sort}[2] = \text{„car“}$, $\text{Sort}[3] = \text{„rca“}$.

1.3. that, for a given search string P , computes and outputs the first and the last index position, where P is a prefix of the rotations stored in Sort .

For example, if $P = \text{„ca“}$, the output should be (2,2)

because 2 is the first index in Sort and the last index in Sort , where Sort contains a rotation that starts with „ca“.

1.4. Extend your program in such a way that it also returns all the index positions in S , where your search string starts.

For a search string „ca“, the index position in S would be 1, because „ca“ starts at position 1 in S .

Here, different implementations are possible.

Try to find an implementation that associates each rotation $R(i)$ with the start position i in S where it starts,

and to extend the data structure for Sort in such a way

that S stores both informations,

i.e. the rotated string, e.g. „car“, and its start position i in S , e.g. 1, together.

As a test, let your program find all start position of the substring „bra“ in $S = \text{„abracadabra“}$.

The output should be the following start positions in S : 2 and 9.

Please send your solution to the programming task by Saturday 19th of September to stb@upb.de with topic „programming task“.

I am looking forward to seeing you on Tuesday 23rd of September,

Stefan Böttcher

