Project 14

Minie Jung, Kunaphas Kongkitimanon, Wan-Ju Chang

Abstract

The goal of the project: Implement two problems of the contest problems.

The main result of the project: We implement task 1 and 7 in C++.

Keywords: suffix array; longest common prefix; string matching

An estimation of the time: 16h

Project evaluation: 4

The number of words: 337 words

Background

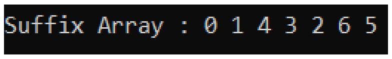
Suffix array is an efficient way which use slight space to sort array of suffixes of a string. We will solve given tasks from applications in contest problems by using the suffix arrays.

Task 1: Hidden Password

Task 1, problem base on This Consider a string of length n (1 <= n <= 100000). Determine its minimum lexicographic rotation. For example, the rotations of the string “alabala” are:
alabala labalaa abalaal balaala alaalab laalaba aalabal
The smallest among them is “aalabal”.

Implementation

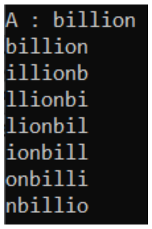
Step1 : Construct suffix array with given string. We use string "billion".



```
Suffix Array : 0 1 4 3 2 6 5
```

Figure 1: Suffix array of string "billion"

Step2 : Show rotation of given string.

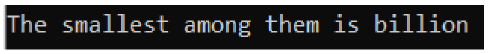


```
A : billion
billion
illionb
llionbi
lionbil
ionbill
onbilli
nbillio
```

Figure 2: Rotation of string "billion"

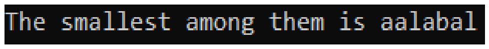
Step3 : Choose minimum lexicographic rotation using suffix array. First element of suffix array is starting position of minumum rotation of given string.

Example Result



```
The smallest among them is billion
```

Figure 3: Result of task 1 (string "billion")



```
The smallest among them is aalabal
```

Figure 4: Result of task 1 (string "alabala")

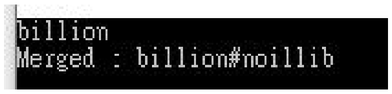
Task 7: The Longest Palindrome

Task 7, the longest palindrome, given a strings S of length n (n <= 20000) determine its longest substring that also is a palindrome. Palindrome is a characters read the same backward as forward, for examples of palindromic words , level, mom, radar, refer, etc.

Implementation

First, we build two variables to store the string (S) we want to analysis and reverse string (RS).

Step1: we combine two together and put '#' in the middle, so it shows like $S + \# + RS$. We use string "billion" to display.

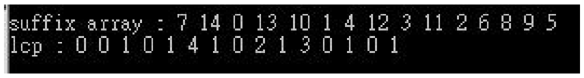


```
billion
Merged : billion#noillib
```

Figure 5

Step2: construct the Suffix Array which is an array of integers giving the starting positions of suffixes of a string in lexicographical order. The array that holds Index of starting position is a suffix Array.

Step3: we compare adjacent suffixes to count the number of longest common prefixes (LCP).

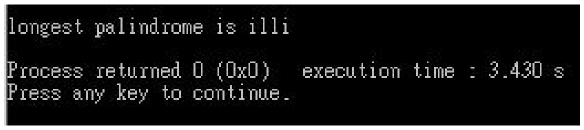


```
suffix array : 7 14 0 13 10 1 4 12 3 11 2 6 8 9 5
lcp : 0 0 1 0 1 4 1 0 2 1 3 0 1 0 1
```

Figure 6: Input string 'billion', construct suffix array. Compare each two of substring to calculate LCP.

Step4: choosing the suffix array which have the longest common prefixes and calculating longest prefixes between suffixArray[i-1] and suffixArray[i]. Thus, we can get the length of longest palindrome.

Example Result



```
longest palindrome is illi

Process returned 0 (0x0)   execution time : 3.430 s
Press any key to continue.
```

Figure 7