CSCI803 Assignment

Yao Xiao
SID 2019180015

November 26, 2020

1 Idea

In fact, the core understanding of the algorithm is to check whether there is a character that satisfies the longest substring condition: the string is displayed as a prefix, suffix, and middle position. He puts the first letter of all strings in the same position at the beginning, because only these positions may have to satisfy the substring condition. Then from front to back, starting from the position, the string that meets condition 1 appears in the middle of the prefix.

In fact, this issue is related to the KMP algorithm. $nxt[i]$ represents the maximum length of the prefix and suffix of the i-th position (after jumping to this position) before and after the i-th position of the same pattern string. Then we can find the next one after the longest common suffix appears in the middle and can be judged, instead of looking at whether there is a secondary suffix in the middle long ago and did the same thing, until we jump to the external string.

So how to determine whether there has been in the middle? We only need to label $nxt[i]$ with the array value that appears on the determined nxt table. Finally, starting from the end position, check whether $nxt[i]$ has marked the value or not, the same suffix length as the middle of the front will be described, not marked in the nxt [$nxt[i]$] double length position, the center mark is not until The mark is found, the record length is exited or the end is not satisfied.

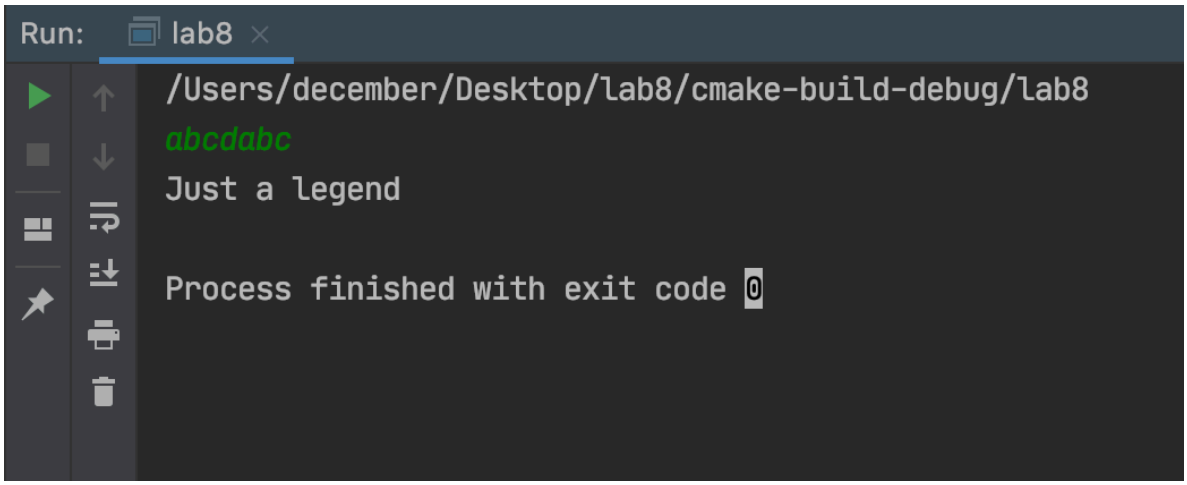
2 Implementation

```
#include <iostream>
#include <string>
#define N 1000002
using namespace std;

string s;
int len;
int nxt[N];
int cnt[N];
int length = 0;
int get_next()
{
    int j = 0;
    int k = -1;
    nxt[0] = -1;
```

```
16      while(j<len)
17      {
18          if(k==-1||s[k]==s[j])
19          {
20              ++k;
21              ++j;
22              nxt[j] = k;
23          }
24          else
25          {
26              k=nxt[k];
27          }
28      }
29      for(int i = 0;i<len;i++)
30      {
31          cnt[nxt[i]] = 1;
32      }
33  }
34
35
36  int main()
37  {
38      cin >> s;
39      len = s.size();
40      get_next();
41      int k = nxt[len];
42      int flag = 0;
43      while(k>0)
44      {
45          if(cnt[k])
46          {
47              flag = 1;
48              length = k;
49              break;
50          }
51          k = nxt[k];
52      }
53      if(flag)
54      {
55          for(int i = len-length;i<len;i++)
56          {
57              cout << s[i];
58          }
59          cout << endl;
60      }
61      else
62      {
63          cout << "Just a legend" << endl;
64      }
65      return 0;
66  }
```


Run:
lab8
/Users/december/Desktop/lab8/cmake-build-debug/lab8
fixprefixsuffix
fix
Process finished with exit code 0


Run:
lab8
/Users/december/Desktop/lab8/cmake-build-debug/lab8
abcdabc
Just a legend
Process finished with exit code 0