صورت سوال

یک رشته‌ی S داده شده است. عدد p را معتبر می‌گوییم اگر برای هر جایگاه i که هر دو جایگاه i و $i+p$ داخل رشته هستند، داشته باشیم $S_i = S_{i+p}$.

همه‌ی مقدارهای معتبر p را پیدا کنید.

الگوریتمی با زمان $O(n)$ ارائه دهید.

برای پیدا کردن تمام مقادیر معتبر p در زمان O(n) ، از الگوریتم Z (یا معادل آن، Prefix Function در KMP) استفاده می‌کنیم.

عدد p زمانی معتبر است که برای تمام اندیس‌های $0 \le i < n-p$ رابطه زیر برقرار باشد:

$S[i] = S[i + p]$

به عبارت دیگر، پیشوند رشته به طول n-p با زیررشته‌ای که از اندیس p شروع می‌شود برابر باشد:

$S[0 \dots n-p-1] = S[p \dots n-1]$

اگر $Z[p]$ طول بزرگ‌ترین پیشوند مشترک رشته و زیررشته شروع‌شده از اندیس p باشد، آنگاه:

pمعتبر است اگر و تنها اگر:

$Z[p] \ge n - p$

الگوریتم

1. آرایه Z را در زمان O(n) محاسبه میشود
2. برای هر مقدار p از 1 تا n-1:
   - اگر $Z[p] \ge n-p$ باشد، pمعتبر است.
3. در صورت نیاز، مقدار p = n نیز معتبر است؛ زیرا هیچ اندیسی برای بررسی باقی نمی‌ماند.

شبه‌کد

```
Z = ZAlgorithm(S)

for p = 1 to n-1:
    if Z[p] >= n-p:
        print(p)

print(n)
```

پیچیدگی زمانی

- محاسبه آرایه Z: O(n)

- بررسی تمام مقادیر p: O(n)

در نتیجه، پیچیدگی زمانی کل الگوریتم برابر با O(n) است.

```cpp
#include <iostream>
#include <vector>
using namespace std;

// Z-function to compute Z-array
vector<int> zFunction(string &s) {
    int n = s.length();
    vector<int> z(n);
    int l = 0, r = 0;

    for (int i = 1; i < n; i++) {
        if (i <= r) {
            int k = i - l;

            // Case 2: reuse the previously computed value
            z[i] = min(r - i + 1, z[k]);
        }

        // Try to extend the Z-box beyond r
        while (i + z[i] < n && s[z[i]] == s[i + z[i]]) {
            z[i]++;
        }

        // Update the [l, r] window if extended
        if (i + z[i] - 1 > r) {
            l = i;
            r = i + z[i] - 1;
        }
    }

    return z;
}
```