پرسش ۱۰ تمرین ۵ **محسن زارع-۴۰۳۱۰۶۰۱۳**

ایده‌ی حل

برای حل این مسئله باید بررسی کنیم که آیا می‌توان رشته‌ی **T** را با یک حرکت به سمت راست و سپس یک حرکت به سمت چپ روی رشته‌ی **S** ساخت یا خیر.

۱-برای اینکه مجبور نباشیم زیررشته‌ها را بارها و بارها به صورت کاراکتر به کاراکتر مقایسه کنیم، ابتدا روی رشته‌ی **S** و همچنین روی معکوس آن **Hash** محاسبه می‌کنیم. به این ترتیب، بعد از یک پیش‌پردازش اولیه، مقایسه‌ی هر زیررشته تنها در زمان **(O(1))** انجام می‌شود.

نکته ی تکمیلی:

پیش‌پردازش هش

ایده این است که برای رشته‌ی S یک هش پیشوندی بسازیم.

برای هر موقعیت i تعریف می‌کنیم:

$$H[i] = H[i-1] \times B + S[i]$$

که در آن:

- Bیک عدد پایه (مثلاً 31 یا 911382323) است .
- $H[i]$هش پیشوند رشته تا اندیس i را نگه می‌دارد .

همچنین توان‌های B را هم از قبل ذخیره می‌کنیم:

$$P[i] = B^i$$

حالا هش هر زیررشته‌ی

$$S[l..r]$$

در زمان ثابت به دست می‌آید:

$$Hash(l,r) = H[r] - H[l-1] \times P[r-l+1]$$

بنابراین بعد از پیش‌پردازش، دیگر لازم نیست کاراکترها را یکی‌یکی مقایسه کنیم.

۲-سپس هر موقعیت از رشته‌ی **T**را به عنوان محل تغییر جهت در نظر می‌گیریم؛ یعنی فرض می‌کنیم تا آن نقطه به سمت راست حرکت کرده‌ایم و از آنجا به بعد به سمت چپ برگشته‌ایم(pivot در نظر میگیریم)

۳-حالا تمام نقاط شروع ممکن در **S** را بررسی می‌کنیم. برای هر نقطه‌ی شروع و هر محل تغییر جهت، دو قسمت رشته‌ی **T** را با بخش‌های متناظر در **S** مقایسه می‌کنیم:

- قسمت اول با حرکت به راست،
- و قسمت دوم با حرکت به چپ.

از آنجا که این مقایسه‌ها با استفاده از Hash انجام می‌شوند، هر بررسی زمان ثابتی دارد. اگر هر دو قسمت با هم تطابق داشته باشند، یعنی رشته‌ی **T** قابل تولید است.

پیچیدگی

پیش‌پردازش Hash برابر **(O(|S|)** است. سپس برای هر نقطه‌ی شروع در **S** و هر محل تغییر جهت در **T** یک بررسی انجام می‌دهیم؛ بنابراین پیچیدگی زمانی الگوریتم برابر است با:

$$O(|S| \times |T|)$$

که در بدترین حالت **(O(n^2)**خواهد بود.