۱. Resequencing / Reference–Based Sequencing

در این مسأله هدف آن است که با وجود داشتن یک ژنوم مرجع و Read هایی که از یک ژنوم هدف خوانده شده‌اند بتوانیم ژنوم هدف را به دست آوریم. در ابتدا بایستی بدانیم که چه شباهت‌ها و تغییراتی بین دو رشته ممکن است وجود داشته باشد ولیکن در حال حاظر بدان نمی‌پردازیم.

یک روش ساده برای بازسازی رشته مرجع آن است که Read ها را روی رشته مرجع تصویر نموده (map) تا بتوانیم اختلاف رشته‌ها را به دست آوریم. اگر تمام تصویرها درست باشند آنگاه به سادگی می‌توان رشته هدف را به دست آورد.

شکل ۱- map کردن read ها به یک ژن reference

در این حالت ابتدا به Read Mapping خواهیم پرداخت و سپس به Calling می‌پردازیم.

۲. مسأله Read Mapping

این مسأله نیازمند دانستن رابطه بین دو رشته بوده و فعلاً درباره آن بحث نمی‌نماییم. در ابتدا به مسأله ساده زیر نگاه می‌کنیم.

۱.۱ Exact Pattern Matching

اگر یک متن T و یک الگوی P که در آن $|T|=n$ و $|P|=m$ می‌باشند را داشته باشیم. می‌خواهیم تمام مکان‌های روی T را پیدا کنیم که P اتفاق افتاده است.

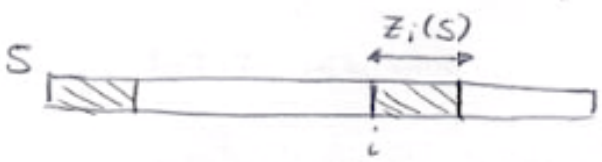
روش ساده برای حل این مسأله آن است که روی متن T یکی‌یکی جلو رفته و ببینیم که آیا P در آن اتفاق افتاده است و یا خیر. این کار نیاز به $O(mn)$ عملیات داشته و با DP هم می‌توان این کار را انجام داد. ولیکن دنبال آن هستیم که این کار را سریعتر انجام دهیم.

نگارنده: سیاوش نظری
ایمیل: sia.vsh92@gmail.com

۱.۲ الگوریتم Z

الگوریتمی ساده که می‌توان الگو را در زمان $O(m+n)$ در رشته پیدا کرد. این الگوریتم بر پایه تعریف تابع زیر می‌باشد.

تعریف تابع Z: برای S و برای $i=۲،۳،\ldots,|S|$ ، طول بلندترین زیررشته S می‌باشد که از مکان i ام شروع گردیده و با یک پیشوند S تطابق دارد $z_i(S)$ می‌نامیم.



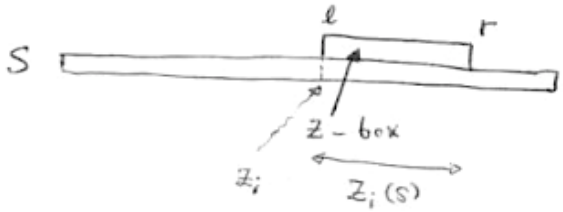
شکل ۱- تابع z

با وجود تابع z به سادگی مسأله را می‌توان حل نمود. تنها کافی است که قرار دهیم $S=P\$T$ که $ حرفی می‌باشد که نه در T و نه P وجود دارد. در این حالت می‌توان نقاطی که $|z_i(S)=m=|P$ می‌باشد را به دست آورد و این نقاط معادل نقاط دلخواه می‌باشد. پیدا کردن این نقاط در $O(n)$ قابل حصول است. بایستی توجه کرد که با وجود $ داریم $z_i(S)\leq m$.

حال سؤال این است که آیا برای یک رشته S می‌توان تابع z را در $O(|S|)$ به دست آورد یا خیر. اگر این امر صورت پذیرد آنگاه مسأله در $O(m+n)$ حل خواهد گردید.

۱.۳ ساختن تابع z در $O(|S|)$

ابتدا یک جعبه z را به عنوان آخرین جعبه‌ای از سمت راست آخرین است که تا به حال یافت گردیده است تعریف نماییم. همین بازه مورد نظر را مورد توجه قرار می‌دهیم.



شکل ۳- یک پنجره z

طبق تعریف $z_i(S)$ می‌دانیم که پنجره فوق در پسوند S تکرار شده است و قابل بسط دادن هم نیست.

حال برای یافتن Z_{i+1} به نقطه مشابه در رشته پسوند یعنی Z_2 نگاه می‌کنیم، چند حالت ممکن است رخ دهد.

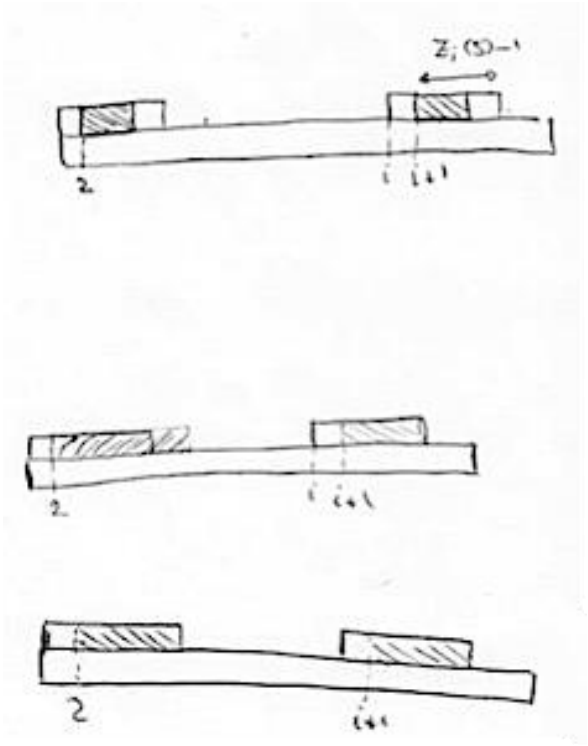
۱. $Z_2(S) < Z_i(S)-1$

در این حالت $Z_2(S) = Z_{i+1}(S)$ می‌باشد.

۲. $Z_2(S) > Z_i(S)-1$

در این حالت $Z_i(S)-1 = Z_{i+1}(S)$ می‌باشد.

۳. $Z_2(S) = Z_i(S)-1$



شکل ۴- حالت‌های مختلف

در این حالت چون حروف آخر با یکدیگر متفاوت می‌باشند بایستی دید که آیا می‌توان از سمت راست پنجره را باز نمود. به همین خاطر نیاز داریم که مکان $Z_i(S)+L$ را با $Z_i(S)+1$ مقایسه نماییم و اگر مساوی بودند ادامه دهیم تا به یک نامساوی برسیم حال $Z_i(S)-1 = Z_{i+1}(S)$ و پنجره را بایستی جدید نماییم.

مراحل ۱ و ۲ را می‌توان پشت هم تکرار نمود. تنها تفاوت آن است که در حالت دوم $Z_j(S)-j+i = Z_{i+1}(S)$ خواهد بود. هرگاه به حالت سوم رسیدیم کار را دوباره آغاز می‌نماییم.

پیچیدگی حافظه $O(|S|)$ می‌باشد دلیل آن اینست که تنها کافی است $Z_i(S)$ ها را نگاه داریم. برای یافتن پیچیدگی زمانی دقت می‌نماییم که برای یافتن $Z_i(S)$ حداکثر یک بار مقایسه‌ای رخ می‌دهد که در آن اختلاف (missmatch) وجود دارد و آن هم

تنها در مرحله سوم صورت می‌پذیرد. حال بایستی دید چند مقایسه وجود دارد که در آن match مشاهده می‌گردد. نکته آنست که هر مقایسه‌ای که با match باشد را یک واحد جلو می‌برد و در نتیجه تعداد آنها حداکثر |S| خواهد بود. بنابراین پیچیدگی زمانی برابر O(|S|) می‌باشد.

بایستی توجه داشت که برای مسأله اصلی که S=P$T می‌باشد کافیست تنها z_m z_{m-1} ... z_2 را نگاه داریم (همه پیشوندها کمتر از m هستند) بنابراین میزان حافظه مورد نیاز برابر است با O(m) .

مشکل این الگوریتم آنست که اگر d الگوی P_d P_{d-1} ... P_1 داشته باشیم آنگاه میزان محاسبات برابر خواهد بود با:

$$O\left(dn + \sum_{i=1}^{d} |P_i|\right)$$

که با مقدار d افزایش می‌یابد و مشکل دیگر آنست که برای مسأله index matching مناسب نمی‌باشد. بنابرین هدف را می‌گذاریم که متن T را با Preprocessing آماده نماییم.