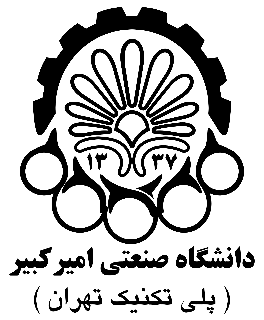
به نام خداوند بخشنده و مهربان



**درس مقدماتی بر بیوانفورماتیک**

گزارش بخش برنامه‌نویسی تمرین دوم

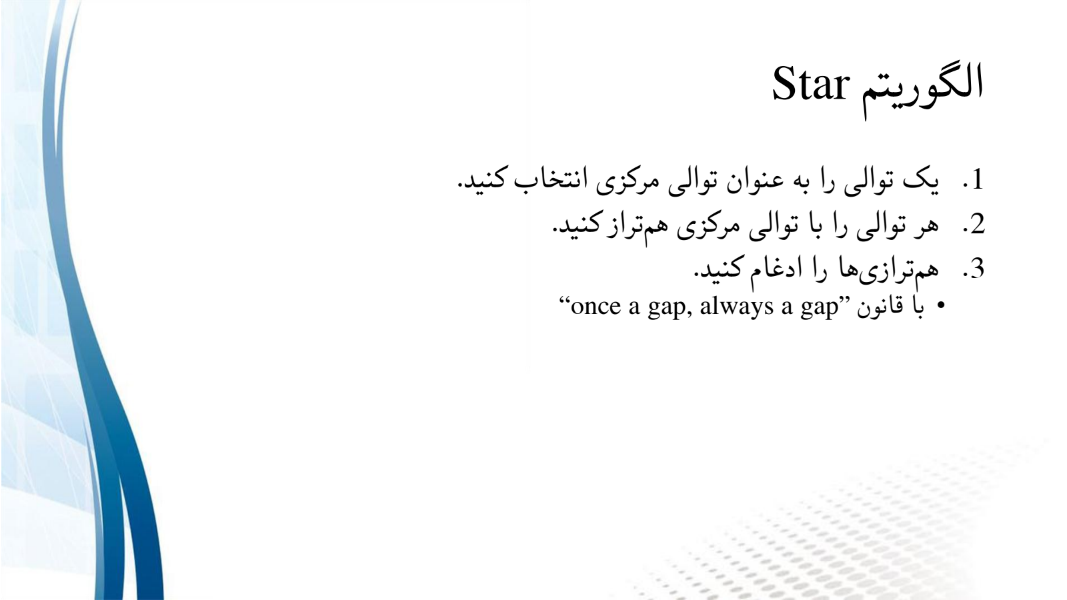
استاد درس: دکتر زینعلی

نام دانشجو:

روزبه قاسمی 9531424

فروردین 1399

**مقدمه**

برای پیاده سازی الگوریتم star باید سه مرحله اصلی آنرا پیاده سازی کرد تا نهایت وقتی چند توالی به برنامه می‌دهیم، مطابق خواسته مسئله و بر اساس امیتازات match، mismatch، gap open و gap extension هم‌ترازسازی‌ها بدست می‌آیند. مراحل آن مطابق اسلاید زیر است:

**مرحله اول الگوریتم Star**

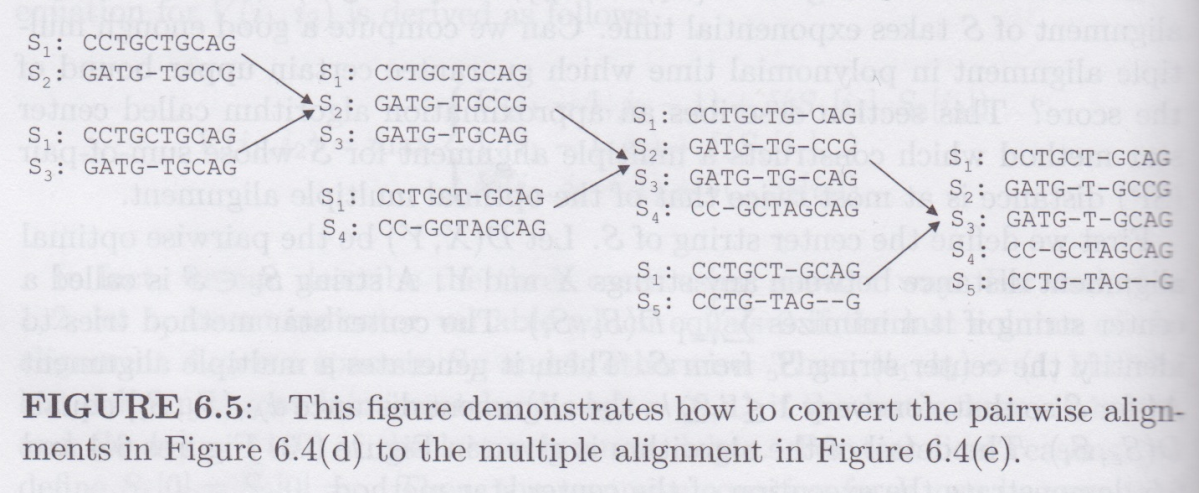
برای پیدا کردن توالی مرکزی باید بر اساس برنامه تمرین قبلی هر زوج رشته را به برنامه قبلی بدهیم تا امتیاز بدست آمد، سپس **ماکزیمم جمع امتیازهای هر سطر، برابر با توالی منتخب ماست.** بر اساس تست کیس داده شده جدول زیر بدست می‌آید(طبق خواسته گفته شده در کانال بر همان اساس امتیازات هر مرحله به دست می‌آید و امتیازات match و ... بر اساس امتیازات گفته شده در تمرین قبل است):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Max | S4 | S3 | S2 | S1 |  |
| -39 | -15 | -13 | -11 |  | S1 |
| -36 | -11 | -14 |  | -11 | S2 |
| -39 | -12 |  | -14 | -13 | S3 |
| -38 |  | -12 | -11 | -15 | S4 |

بنابراین توالی دوم به عنوان توالی مرکزی انتخاب می‌شود.

**مرحله دوم الگوریتم Star**

سپس در این مرحله باید دوبه‌دو مطابق الگوریتم زیر باهم ادغام شود، بر این اساس که همانند مثالی که جدول آن را در مرحله اول به دست آوردیم، با هم همتراز می‌کنیم:

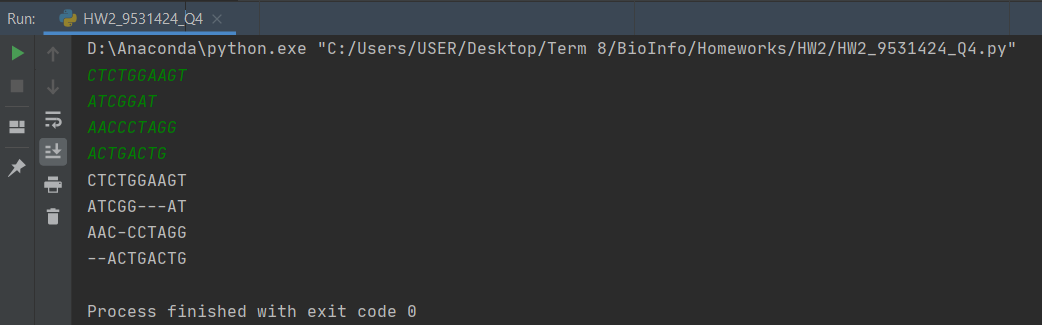


**مرحله سوم الگوریتم Star**

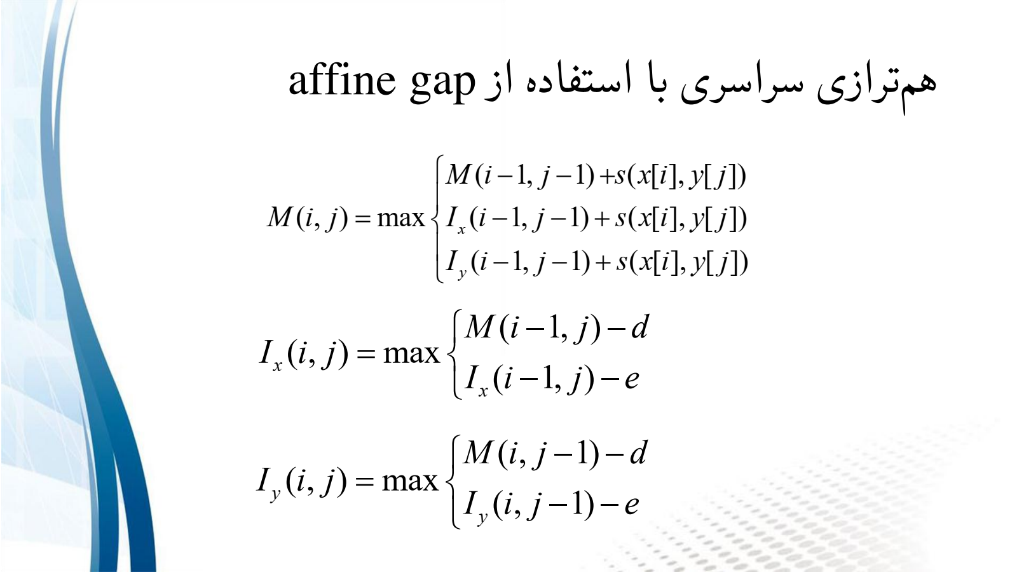
سپس آن‌ را باهم ادغام می‌کنیم:

if sum\_list[1] == max\_val:  
 # print("The Sc is second sequence: ")  
  
 res\_seq\_0\_2 = Affine\_gap(GA\_second\_seq\_1\_0, GA\_second\_seq\_1\_2)  
 GA\_second\_seq\_1\_0, GA\_second\_seq\_1\_2, num\_seq\_1\_0 = res\_seq\_0\_2.alingment()  
 res\_seq\_0\_2 = Affine\_gap(GA\_second\_seq\_1\_0, GA\_second\_seq\_1\_3)  
 GA\_second\_seq\_1\_0, GA\_second\_seq\_1\_3, num\_seq\_1\_0 = res\_seq\_0\_2.alingment()  
 res\_seq\_0\_2 = Affine\_gap(GA\_second\_seq\_1\_2, GA\_second\_seq\_1\_3)  
 GA\_second\_seq\_1\_2, GA\_second\_seq\_1\_3, num\_seq\_1\_0 = res\_seq\_0\_2.alingment()  
 res\_seq\_0\_2 = Affine\_gap(GA\_first\_seq\_1\_0, GA\_second\_seq\_1\_3)  
 GA\_first\_seq\_1\_0, GA\_second\_seq\_0\_3, num\_seq\_1\_0 = res\_seq\_0\_2.alingment()  
  
 print(GA\_second\_seq\_1\_0)  
 print(GA\_first\_seq\_1\_0)  
 print(GA\_second\_seq\_1\_2)  
 print(GA\_second\_seq\_1\_3)

**تست برنامه**

خروجی تست برنامه آن در نهایت به شکل زیر است:

**چالش های پیاده‌سازی**

بر اساس خروجی هایی که برای تست‌های مطرح شده در کوئرا قرار دادید، خروجی من با خروجی جواب یکی نیست و نمره 0 به من تعلق گرفته است. این درحالی است که مشکل نه از پیاده سازی من است و نه از خروجی شما بلکه علت آن باز می‌گردد به قسمت Alignment کردن تمرین قبلی چون که اگر توجه نمایید، برای Traceback کردن ما مطابق اسلاید زیر عمل می‌کنیم:

حال اگر دقت کنید متوجه می‌شوید که به طور مثال برای ماتریس M وقتی که ما در زمان بدست آوردن ماکزیمم، بیشتر از یکی از آنها درست شود، باعث می‌شود که شرط if یا elseif شود دیگر بقیه چک نمی‌شود و مشکل از همینجاست!

تنها راه حلی که به ذهن من می‌رسید این بود که هر 12 حالت( 3 حالت برای M، 2 حالت برای Ix و 2 حالت برای Iy پس در نهایت 2\*2\*3 حالت)، باید بررسی شود سپس انتخاب کنیم که کدام هم‌ترازسازی مطابق میل ماست. اما از آنجایی که این روش بسیار روش Inefficient ای به حساب می‌آید زیرا بسیار حل مسئله آن در این حالت زمان‌بر خواهد بود و راه دیگری به ذهنم نمی‌رسید برای همین هم‌ترازسازی هایی که من بدست آوردم، درست است و امتیاز آن نیز به درستی به‌دست آمده اما یکی از حالت هایی است که شما آنرا در نظر نگرفته اید. بنابراین پیاده سازی من کاملاً درست و منطقی است اما از لحاظ کوئرا نمره ای به آن تعلق نمی‌گیرد!

پایان