پروژه بیوانفورماتیک - فاز اول

فاطمه توکلی
علی صفرپور دهکردی
سید صالح اعتمادی
دانشگاه علم و صنعت ۹۸-۹۷

لطفا به نکات زیر توجه کنید:

- مهلت ارسال این تمرین شنبه ۲۲ تیر ماه ساعت ۱۱:۵۹ ب.ظ است.
- این تمرین شامل سوال های برنامه نویسی می باشد، بنابراین توجه کنید که حتماً موارد خواسته شده را رعایت کنید. .
- نام شاخه، پوشه و پول ریکوست همگی دقیقا $AG1$ باشد.
- اگر در حل پروژه بیوانفورماتیک مشکلی داشتید،لطفا به $@S\_Aliiiii$ یا $@Fateme\_tavakoliii$ مراجعه کنید.
-

موفق باشید.

توضیحات کلی تمرین

تمرین این هفته ی شما، ۲ سوال دارد که باید به همه ی این سوال ها پاسخ دهید. برای حل این سری از تمرین ها مراحل زیر را انجام دهید:

۱. ابتدا مانند تمرین های قبل، یک پروژه به نام AG۱ بسازید.

۲. کلاس هر سوال را به پروژه ی خود اضافه کنید و در قسمت مربوطه کد خود را بنویسید. هر کلاس شامل دو متد اصلی است:

متد اول: تابع solve است که شما باید الگوریتم خود را برای حل سوال در این متد پیاده سازی کنید.

متد دوم: تابع process است که مانند تمرین های قبلی در TestCommon پیاده سازی شده است. بنابراین با خیال راحت سوال را حل کنید و نگران تابع process نباشید! زیرا تمامی پیاده سازی ها برای شما انجام شده است و نیازی نیست که شما کدی برای آن بزنید.

۳. اگر برای حل سوالی نیاز به تابع های کمکی دارید؛ می توانید در کلاس مربوط به همان سوال تابع تان را اضافه کنید.

اکنون که پیاده سازی شما به پایان رسیده است، نوبت به تست برنامه می رسد. مراحل زیر را انجام دهید.

۱. یک UnitTest برای پروژه ی خود بسازید.

۲. فولدر TestData که در ضمیمه همین فایل قرار دارد را به پروژه ی تست خود اضافه کنید.

۳. فایل GradedTests.cs را به پروژه ی تستی که ساخته اید اضافه کنید. برای این تمرین مانند $A7$ و $A8$ برای هر سوال تست جداگانه با زمان جداگانه در نظر گرفته شده. بعد از حل کردن هر تمرین Assert.Inconclusive را از ابتدای تست حذف کرده و آن را اجرا کنید. چنانچه سوالی را حل نکردید از Assert.Inconclusive استفاده کنید.

دقت کنید که TestCommon تغییر یافته است. بنابراین شما باید نسخه ی جدید آن را با دستور Pull

git دریافت کنید .

۱ اسمبل کردن ژنوم $phiX174$ از دیتای بدون خطا

در سال ۲۰۱۱ تعداد زیادی از مردم آلمان دچار بیماری واگیر داری شدند. این بیماری که نشانه هایش با مسمومیت غذایی شروع شد و به بستری شدن و یا مرگ می انجامید کم کم در سرتاسر اروپا گسترش پیدا کرد. دولت آلمان علت شیوع این بیماری را از خیار دانست و در نتیجه تعداد زیادی از مزارع خیار را از بین برد. این اشتباه باعث شد تا دولت مجبور به پرداخت میلیون ها دلار خسارت به کشاورزان شود. بعد از تحقیقات گسترده منبع اصلی بیماری جوانه ی لوبیا شناخته شد که از اسپانیا وارد شده بود و در اکثر رستوران ها استفاده میشد. این تحقیقات ادامه داشت تا اینکه نشانه های بیماری در دختری پیدا شد که تست ها، بیمار بودن او را رد میکرد. محققان به این نتیجه رسیدند که آنها با یک ویروس جدید که از قبل شناخته نشده مواجه اند و راه های سنتی جواب نمیدهد محاسبات زیستی لازم است برای اینکه ساختار ویروس را پیدا کنند.

در این سری تمارین شما با همان مسئله ای که دانشمندان آن زمان روبه رو بودند(سر هم کردن ژنوم کشنده ی $E.coliX$)،روبه رو میشوید و قدم هایی که آنها در راین راه گذاشتند را طی میکنید.

بزرگترین مشکل در راه زیست شناسان وجود نداشتن تکنولوژی لازم برای خواندن نوکلئوتید های یک ژنوم (کوچکترین قسمت هایی که یک ژنوم را میسازند) از ابتدا تا انها مانند یک کتاب است(هزینه ی این کار بسیار زیاد و غیر عملی است). بهترین کاری که میتوانند بکنند سر هم کردن تکه های کوچکتری از DNA است . این تکه ها read نام دارند. مشکل سر هم کردن read ها این است که زیست شناسان نمیتوانند بفهمد هر کدام مربوط به کدام قسمت از ژنوم است،آنها بایداز read هایی که قسمت های مشترک دارند ا ستفاده کنند و آنها را سر هم کنند. به طور کلی مسئله ی سر هم کردن ژنوم پیچیده و گسترده است به همین علت ما آنرا به قسمت های ساده تر تبدیل میکنیم تا شما توانایی لازم را برای مسئله اصلی کسب کنید. شما در ابتدا قدم های Fred Sanger را دنبای خواهید کرد. او کسی است که با اسمبل کردن ژنوم $phiX174$ جایزه ی نوبل را در سال ۱۹۷۷ از آن خود کرد. که باعث شد این ویروس به یک ویروس معروف و پایه ای در دنیای ژنومیک تبدیل شود.

$phiX174$ ژنوم دایره است به این معنا که ابتدا و انتهایش یکی است. این ویروس ساده که به تنهایی نمیتواند کسترش پیدا کند و از طرق چسبیدن به باکتری ها و انتقال DNA اش به باکتری زیاد میشود،طولش فقط $5million-nucleotid$ است که در مقایسه با $E.coliX$ که طولش $5,386-nucleotide$ است خیلی کوچکتر میباشد. ما به شما تقریبا ۱۰۰۰ read میدهیم که ارور ندارند. و هر read طولش $100-nucleotid$ است. برای مثال read های زیر را در نظر بگیریدو ببینید چگونه میتوان ساختار ژنوم را از read های DNA به دست آورد.

```
......AGAATATCA
.....GAGAATATC
....TGAGAATAT
```

```
....TGAGAATATCA...
```

read های مثال بالا $9-nucleotid$ هستند ولی ژنوم نهایی میتواند خیلی بلند تر از هر قسمتش (read) باشد .

علت اینکه read ها overlap دارند این است که تعداد زیادی از DNA های یکسان تکه تکه شده اند .

فرمت ورودی: در هر خط read های ژنوم به شما داده میشود دقت کنید که read ها به ترتیب نیستند و این وظیفه ی شما است که با توجه به overlap داده ها ژنوم کامل را پیدا کنید.

هر کدام از ۱۶۱۲ خط ورودی یک read است که یک جمله شامل حروف A, C, G, T میباشد. طول هر read $100-nucleotid$ است به یاد داشته باشید که ممکن است بعضی از read $(100-mer)$ها گم شده باشند و در ورودی داده نشده باشند

مثال)

```
AAC
ACG
GAA
GTT
TCG
```

```
AACGAAGTTCG
```

توضیح خروجی :

در اینجا ژنوم دایره ای $AACGAAGTTCG$ است.و همه ی read ها از ژنوم تشکیل شده اند.

*AACGAA**GTT**CG*، *AAC**GAA**GTTCG*، *A**ACG**AAGTTCG*، ***AAC**GAAGTTCG*،
*AACGAAGT**TCG***،

راهنمایی :

برای جملات ورودی گراف overlap تشکیل دهید. دو راس گراف توسط یک یال وزن دار که وزنش ماکسیمم طول overlap است متصل میشوند. سپس به صورت greedy دور همیلتونی تشکیل دهید توجه داشته باشید که اولویت انتخاب با آن read ای است که در ورودی زودتر آمده است

۲ اسمبل کردن ژنوم $phiX174$ از دیتای مستعد خطا

به شما لیستی از read های ژنوم داده شده است به صورتی که ممکن است خطا در آنها وجود داشته باشد. شما باید ژنوم دایره ای را که این read ها از آن ژنوم ساخته شده اند برگردانید.

فرمت ورودی:

هر کدام از۱۶۱۲خط ورودی شامل یک read است که طول هر کدام $nucleotid - 100$ است . هر کدام از read ها میتواند شامل یک خطا باشد (برای مثال: یکی از حروف در هر read میتواند اشتباه باشد ولی حذف یا اضافه نمیشوند) هر ناحیه ی overlap هم میتواند حداکثر ۱۰٪ خطا داشته باشد. علت وجود خطا این است که در واقعیت نیز ماشین های مخصوص اسمبل کردن ژنوم ها ۱٪ خطا دارن در واقع فرض براین است که براثر خوانده شدن اطلاعات بسیار زیاد احتمال خطا کتمان ناپذیر است؛همچنین با توجه به اینکه در یک ناحیه اشتراکی که معقول نمی باشد که حرف های تغییر کرده بیش از حد متراکم شوند،در بیان کلی و با توجه به حساسیت رشته ها با تغییر بیش از حد مجاز ، عملا رشته را تغییر داده ایم و اطلاعات خروجی مفید نخواهد بود.د. بقیه ی فرض ها مانند سوال قسسمت قبل است و محدوده ی حروف همان A،T،C،G میباشد

مثال ۱)(بدون خطا)

```
AAC
ACG
GAA
GTT
TCG
```

```
AACGAAGTTCG
```

نکته:

ما در مثال بالا خطا قرار ندادیم زیرا طول read ها در مقایسه با حالت عادی خیلی کوتاه است و اگر خطا داشت ، سر هم کردنش خیلی سخت میشد.