

Semi-Global alignment

- محدودیت زمان: 1.5 ثانیه
- محدودیت حافظه: 100 مگابایت

در این تمرین قرار است با زبان پایتون، Semi-Global alignment را برای پروتئین‌ها پیاده‌سازی کنید (با متد Dynamic Programming). به این شکل که تمام Gap های اول و آخر alignment را در امتیاز دهی در نظر نگیرید. همچنین در این تمرین لازم است تمام جواب های ممکن را در TraceBack بدست آورید.

برای Match و Mismatch از ماتریس PAM250 استفاده کنید و هزینه gap را هم 9 در نظر بگیرید.

برای راحتی شما دیکشنری PAM250 در زیر آورده شده که کافی است آن را در ابتدای کد خود copy/paste کنید:

```
PAM250 = {
'A': {'A': 2, 'C': -2, 'D': 0, 'E': 0, 'F': -3, 'G': 1, 'H': -1, 'I': -1, 'K': -1,
'C': {'A': -2, 'C': 12, 'D': -5, 'E': -5, 'F': -4, 'G': -3, 'H': -3, 'I': -2, 'K': -5,
'D': {'A': 0, 'C': -5, 'D': 4, 'E': 3, 'F': -6, 'G': 1, 'H': 1, 'I': -2, 'K': 0,
'E': {'A': 0, 'C': -5, 'D': 3, 'E': 4, 'F': -5, 'G': 0, 'H': 1, 'I': -2, 'K': 0,
'F': {'A': -3, 'C': -4, 'D': -6, 'E': -5, 'F': 9, 'G': -5, 'H': -2, 'I': 1, 'K': -5,
'G': {'A': 1, 'C': -3, 'D': 1, 'E': 0, 'F': -5, 'G': 5, 'H': -2, 'I': -3, 'K': -2,
'H': {'A': -1, 'C': -3, 'D': 1, 'E': 1, 'F': -2, 'G': -2, 'H': 6, 'I': -2, 'K': 0,
'I': {'A': -1, 'C': -2, 'D': -2, 'E': -2, 'F': 1, 'G': -3, 'H': -2, 'I': 5, 'K': -2,
'K': {'A': -1, 'C': -5, 'D': 0, 'E': 0, 'F': -5, 'G': -2, 'H': 0, 'I': -2, 'K': 5,
'L': {'A': -2, 'C': -6, 'D': -4, 'E': -3, 'F': 2, 'G': -4, 'H': -2, 'I': 2, 'K': -3,
'M': {'A': -1, 'C': -5, 'D': -3, 'E': -2, 'F': 0, 'G': -3, 'H': -2, 'I': 2, 'K': 0,
'N': {'A': 0, 'C': -4, 'D': 2, 'E': 1, 'F': -3, 'G': 0, 'H': 2, 'I': -2, 'K': 1,
'P': {'A': 1, 'C': -3, 'D': -1, 'E': -1, 'F': -5, 'G': 0, 'H': 0, 'I': -2, 'K': -1,
'Q': {'A': 0, 'C': -5, 'D': 2, 'E': 2, 'F': -5, 'G': -1, 'H': 3, 'I': -2, 'K': 1,
'R': {'A': -2, 'C': -4, 'D': -1, 'E': -1, 'F': -4, 'G': -3, 'H': 2, 'I': -2, 'K': 3,
'S': {'A': 1, 'C': 0, 'D': 0, 'E': 0, 'F': -3, 'G': 1, 'H': -1, 'I': -1, 'K': 0,
'T': {'A': 1, 'C': -2, 'D': 0, 'E': 0, 'F': -3, 'G': 0, 'H': -1, 'I': 0, 'K': 0,
'V': {'A': 0, 'C': -2, 'D': -2, 'E': -2, 'F': -1, 'G': -1, 'H': -2, 'I': 4, 'K': -2,
'W': {'A': -6, 'C': -8, 'D': -7, 'E': -7, 'F': 0, 'G': -7, 'H': -3, 'I': -5, 'K': -3,
```

```
'Y': {'A': -3, 'C': 0, 'D': -4, 'E': -4, 'F': 7, 'G': -5, 'H': 0, 'I': -1, 'K': -4,}
```

دقت داشته باشید که برنامه ی شما توسط judge کوثر به صورت اتوماتیک بررسی می شود برای همین لازم هست تا جواب های مسئله را دقیقا به ترتیبی که من وارد کردم، بدست بیارید. برای این کار میتوانید از کد زیر برای چاپ جوابا هایتان استفاده کنید.

```
print(score)
sortedSeq = [i[0]+i[1] for i in seq]
sortedSeq.sort()
for i in sortedSeq:
    print(i[0:int(len(i)/2)])
    print(i[int(len(i)/2):])
```

در این تیکه کد، score امتیاز alignment است و seq تمام حالت های ممکن alignment است که به این شکل است.

```
seq = [('AAAAA', '---AA'), ('AAAAA', 'AA---'), ('AAAAA', '-AA--'), ('AAAAA', '--AA-')]
```

این نمونه مقدار مربوط به مثال دو است.

ورودی

ورودی شامل دو string با حروف بزرگ است که در دو خط متفاوت داده می شوند و هر کدام نشان دهنده ی یکی از پروتیین ه جهت مقایسه است و طول آنها هم حداکثر 100 کراکتر است.

خروجی

در خط اول امتیاز alignment خود را چاپ کنید و در خطوط بعدی تمام حالت های ممکن برای alignment را همانند مثال ها چاپ کنید. برای Gap از - استفاده کنید و ترتیب را هم با استفاده از شبه کد بالا رعایت

کنید.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده شده است.

ورودی نمونه ۱

HEAGAWGHE
PAWHEA

خروجی نمونه ۱

20
HEAGAWGHE-
---PAW-HEA

ورودی نمونه ۲

AAAAA
AA

خروجی نمونه ۲

4
AAAAA
---AA
AAAAA
--AA-
AAAAA
-AA--

AAAAA

AA---

- دقت کنید که مثال دو، چهار جواب متفاوت دارد که هر چهار جواب را ترتیب ذکر شده در شبه کد بالا در خروجی قرار گرفته اند.