

Points: 120

Prefix alignment

Write a Python program that will do a prefix alignment. A prefix alignment finds the best match starting at the beginning of both strings and using gap penalties as in global alignment but assumes no gap penalty for gaps at the end of South string. This is looking for South strings that match the first part of the East string. Another way to think of this is South strings are potential prefixes for a given East string. This could be used to look for them.

For this assignment, assume that in cases of equality a substitution will be taken before an South string gap before a East gap. Input will be in the format shown in class. First a list of the possible characters separated by spaces; then the scoring matrix; then the gap penalty; then the East string and South string. Use a dictionary for storing the substitution matrix. Example input looks like:

```
e a t o n r i
5 -2 -3 -2 -3 -3 -1
-2 5 -3 -2 -3 -3 -2
-3 -3 5 -3 -3 -3 -3
-2 -2 -3 5 -3 -3 -2
-3 -3 -3 -3 5 -3 -3
-3 -3 -3 -3 -3 5 -3
-1 -2 -3 -2 -3 -3 5
-7
orientation
orient
```

The output will look like:

```
East: orientation
South: orient
```

The directions:

```
. - - - - - - - - - -
| \ - - - - - - - - \ -
| | \ - - - - - - - - - -
| | | \ - - - - - - - - - -
| | | | \ - - - - - - - - \
| | | | | \ - - - - - - - - \
| | | | | | \ - \ - - - -
```

The scores:

```
0.0 -7.0 -14.0 -21.0 -28.0 -35.0 -42.0 -49.0 -56.0 -63.0 -70.0 -77.0
-7.0 5.0 -2.0 -9.0 -16.0 -23.0 -30.0 -37.0 -44.0 -51.0 -58.0 -65.0
-14.0 -2.0 10.0 3.0 -4.0 -11.0 -18.0 -25.0 -32.0 -39.0 -46.0 -53.0
-21.0 -9.0 3.0 15.0 8.0 1.0 -6.0 -13.0 -20.0 -27.0 -34.0 -41.0
-28.0 -16.0 -4.0 8.0 20.0 13.0 6.0 -1.0 -8.0 -15.0 -22.0 -29.0
-35.0 -23.0 -11.0 1.0 13.0 25.0 18.0 11.0 4.0 -3.0 -10.0 -17.0
-42.0 -30.0 -18.0 -6.0 6.0 18.0 30.0 23.0 16.0 9.0 2.0 -5.0
```

Score: 30.0

Alignment:

orient
orient

Another example is this one using the same scoring matrix:

East: tartartar
South: tatterer

The directions:

```
. - - - - - - - - -  
| \ - - \ - - \ - -  
| | \ - - \ - - \ - -  
| \ | \ \ - - \ - -  
| \ | \ \ \ - - \ - -  
| | | \ | | \ - - \ - -  
| | | | \ | | \ - - \ - -  
| | | | \ \ | \ \ - - \ - -  
| | | | \ \ | \ \ \ - - \ - -
```

The scores:

```
0.0 -7.0 -14.0 -21.0 -28.0 -35.0 -42.0 -49.0 -56.0 -63.0  
-7.0 5.0 -2.0 -9.0 -16.0 -23.0 -30.0 -37.0 -44.0 -51.0  
-14.0 -2.0 10.0 3.0 -4.0 -11.0 -18.0 -25.0 -32.0 -39.0  
-21.0 -9.0 3.0 7.0 8.0 1.0 -6.0 -13.0 -20.0 -27.0  
-28.0 -16.0 -4.0 0.0 12.0 5.0 -2.0 -1.0 -8.0 -15.0  
-35.0 -23.0 -11.0 -7.0 5.0 10.0 3.0 -4.0 -3.0 -10.0  
-42.0 -30.0 -18.0 -6.0 -2.0 3.0 15.0 8.0 1.0 2.0  
-49.0 -37.0 -25.0 -13.0 -9.0 -4.0 8.0 12.0 6.0 -1.0  
-56.0 -44.0 -32.0 -20.0 -16.0 -11.0 1.0 5.0 9.0 11.0
```

Score: 11.0
Alignment:
tartartar
tatter-er

Submission

As always, do your own work. Do not copy from others or from the internet. Using what we did in class is perfectly legal. Submit a single file named `align.py` to the class submission page. The test script will ignore the spacing you choose and focus on non-whitespace. Remember no late papers and so always turn something in for partial credit.