Programming Homework 3 Instructions

We saw how to adapt dynamic programming to find approximate occurrences of a pattern in a text. Recall that:

* Rows of the dynamic programming matrix are labeled with bases from P and columns with bases from T
* Elements in the **first row are set to 0**
* Elements in the first column are set to 0, 1, 2, ..., as for edit distance
* Other elements are set in the same way as elements of a standard edit distance matrix
* The **minimal value in the bottom row** is the edit distance of the closest match between P and T

First, download the provided excerpt of human chromosome 1

<https://d28rh4a8wq0iu5.cloudfront.net/ads1/data/chr1.GRCh38.excerpt.fasta>

Second, parse it using the readGenome function we wrote before.

Third, adapt the editDistance function we saw in practical (copied below) to answer questions 1 and 2 below. Your function should take arguments p (pattern), t (text) and should return the edit distance of the match between P and T with the fewest edits.

def editDistance(x, y):

# Create distance matrix

D = []

for i in range(len(x)+1):

D.append([0]\*(len(y)+1))

# Initialize first row and column of matrix

for i in range(len(x)+1):

D[i][0] = i

for i in range(len(y)+1):

D[0][i] = i

# Fill in the rest of the matrix

for i in range(1, len(x)+1):

for j in range(1, len(y)+1):

distHor = D[i][j-1] + 1

distVer = D[i-1][j] + 1

if x[i-1] == y[j-1]:

distDiag = D[i-1][j-1]

else:

distDiag = D[i-1][j-1] + 1

D[i][j] = min(distHor, distVer, distDiag)

# Edit distance is the value in the bottom right corner of the matrix

return D[-1][-1]

Hint: In the "A new solution to approximate matching" video we saw that the best approximate match of P =GCGTATGCwithin T =TATTGGCTATACGGTThad 2 edits. You can use this and other small examples to double-check that your function is working.