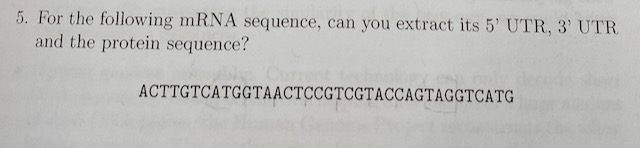
**CSCI 451/551 Fall 2023 Homework 1  
Due: Thu, Sept 7, start of class**

Benjamin Haedt

12 September 2023

**Problem 1 (5 pts)** Exercise 5 (textbook section 1.9, pg 27)



***Clarification:*** this is actually a DNA sequence; you should look for the DNA versions of start and stop codons rather than the RNA version.

Answer: The Start codon is ATG. The stop codon is TAG. Starting from ATG but not including ATG, we convert each three characters into the DNA sequence to RNA, then its protein sequence using <https://biomodel.uah.es/en/lab/cybertory/analysis/trans.htm>. The Protein sequence is VTPQYQ

**Problem 2 (10 pts)** Implement the *Z-algorithm* in the language of your choice. Also, code up the *Exact Pattern Matching* algorithm that makes use of the Z algorithm.  
Demonstrate both algorithms on some test strings (use screen shots to show runs). Include your source code at the end of the PDF.

A screen shot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

\*Code\*

def test():

    string = "caabxaaabaaab"

    pattern = "aab"

    conStr = pattern + "#" +  string

    strlength = len(conStr)

    z = [0] \* strlength

    print(conStr)

    print(strlength)

    l = 0

    r = 0

    for index in range (strlength):

        if index > r:

            l,r = index,index

            while r < strlength and conStr[r-l]==conStr[r]:

                r+=1

            z[index] = r -l

            r-=1

        else:

            k = index - l

            if z[k] < r - index + 1:

                z[index] = z[k]

            else:

                l = index

                while r < strlength and conStr[r-l] == conStr[r]:

                    r+=1

                z[index] = r - l

                r-=1

    for i in range(0, strlength-1):

        if z[i] == len(pattern):

            print(i - len(pattern) - 2)

def book():

    string = "abcdeaabdfaabbfg"

    k = 1

    pattern = "aabb"

    conStr = pattern + "#" +  string

    strlength = len(conStr)

    patL = len(pattern)

    z = [0] \* strlength

    print(strlength)

    l = 0

    r = 0

    for i in range (strlength):

        if i > r:

            r=i

            l= r

            while conStr[r-l] == conStr[r] and r < strlength:

                r+=1

            z[i] = r-l

            r-=1

        else:

            k = i - l

            if z[k] < r - i + 1:

                z[i] = z[k]

            else:

                l = i

                while r < strlength and conStr[r-l] == conStr[r]:

                    r+=1

                z[i] = r - l

                r-=1

    print("Concatenated Pattern and sequence to match: ", string)

    print("Pattern Length: ", str(patL))

    for i in range(strlength):

        if z[i] == patL:

            print("Pattern found at", i - patL - 2)

def main():

    book()

main()