**Labsheet 8**

**KMP**

def KMPSearch(pat, txt):

    M = len(pat)

    N = len(txt)

    lps = [0]\*M

    j = 0

    computeLPSArray(pat, M, lps)

    i = 0

    while (N - i) >= (M - j):

        if pat[j] == txt[i]:

            i += 1

            j += 1

        if j == M:

            print("Found pattern at index " + str(i-j))

            j = lps[j-1]

        elif i < N and pat[j] != txt[i]:

            if j != 0:

                j = lps[j-1]

            else:

                i += 1

def computeLPSArray(pat, M, lps):

    len = 0

    lps[0] = 0

    i = 1

    while i < M:

        if pat[i] == pat[len]:

            len += 1

            lps[i] = len

            i += 1

        else:

            if len != 0:

                len = lps[len-1]

            else:

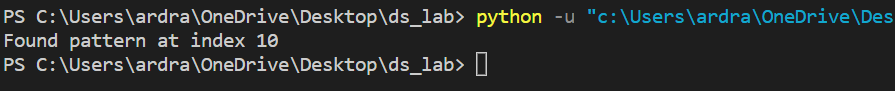
                lps[i] = 0

                i += 1

txt = "ABABDABACDABABCABAB"

pat = "ABABCABAB"

KMPSearch(pat, txt)

****

**Rabin Carp**

d = 256

def search(pat, txt, q):

    M = len(pat)

    N = len(txt)

    i = 0

    j = 0

    p = 0

    t = 0

    h = 1

    for i in range(M-1):

        h = (h\*d) % q

    for i in range(M):

        p = (d\*p + ord(pat[i])) % q

        t = (d\*t + ord(txt[i])) % q

    for i in range(N-M+1):

        if p == t:

            for j in range(M):

                if txt[i+j] != pat[j]:

                    break

                else:

                    j += 1

            if j == M:

                print("Pattern found at index " + str(i))

        if i < N-M:

            t = (d\*(t-ord(txt[i])\*h) + ord(txt[i+M])) % q

            if t < 0:

                t = t+q

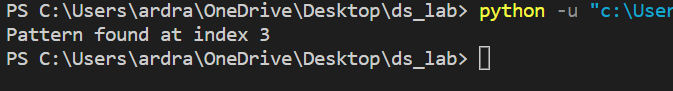
if \_\_name\_\_ == '\_\_main\_\_':

    txt = "ABCCDDAEFG"

    pat = "CDD"

    q = 101

    search(pat, txt, q)

****

**Z Algorithm**

def getZarr(string, z):

    n = len(string)

    l, r, k = 0, 0, 0

    for i in range(1, n):

        if i > r:

            l, r = i, i

            while r < n and string[r - l] == string[r]:

                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 < n and string[r - l] == string[r]:

                    r += 1

                z[i] = r - l

                r -= 1

def search(text, pattern):

    concat = pattern + "$" + text

    l = len(concat)

    z = [0] \* l

    getZarr(concat, z)

    for i in range(l):

        if z[i] == len(pattern):

            print("Pattern found at index",

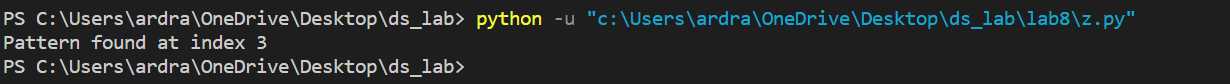
                    i - len(pattern) - 1)

if \_\_name\_\_ == "\_\_main\_\_":

    text = "ABCCDDAEFG"

    pattern = "CDD"

    search(text, pattern)

****