

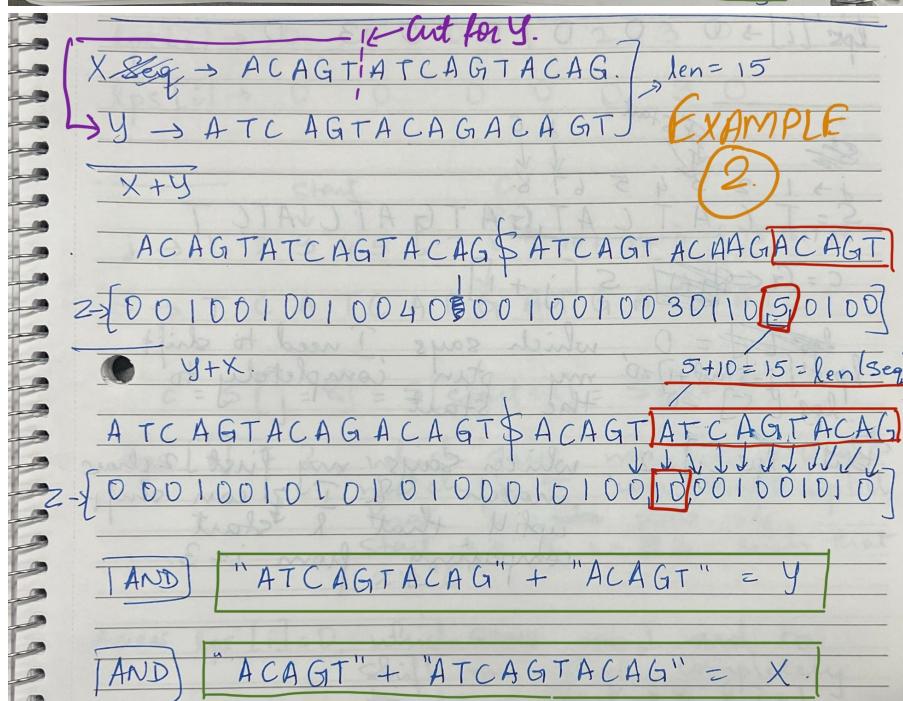
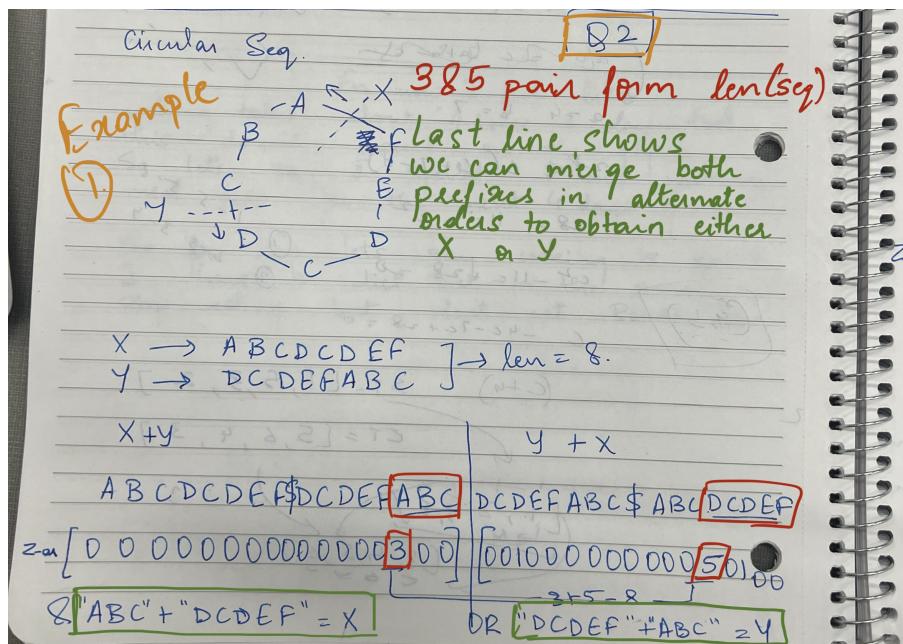
CSE 559 Assign 1

Q2 Solution

Given two linear sequences, X and Y, we can use following steps to show if they are from the same circular sequence

Steps:

1. We can find z-array for both $X + \text{}$ & $Y + \text{}$
 2. We can search for a pair of values, one from each of the z-array, such that, their addition is equal to `len(sequence)` and combining those 2 prefixes results in either of the sequences, X & Y .



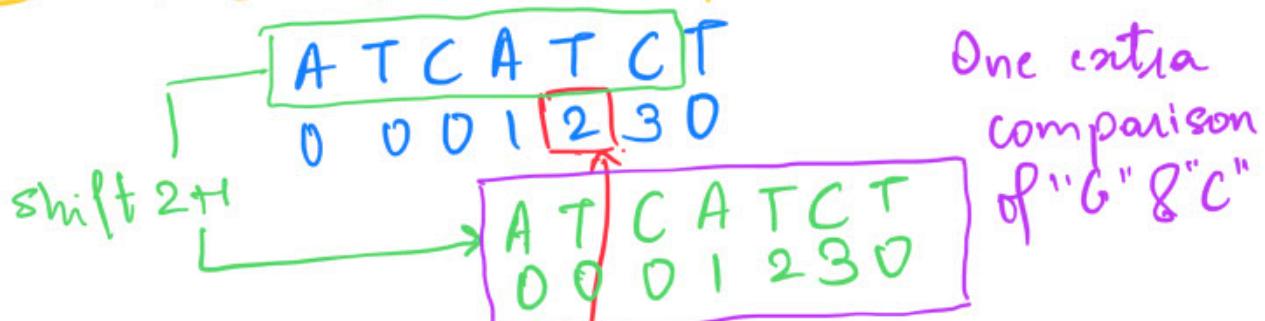
Q3 Solution

$i \rightarrow 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$
 $P[i] \rightarrow A \ T \ C \ A \ T \ C \ T$
 $LPS[i] \rightarrow 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 0$

Step ① LPS

C
↓

$S \rightarrow T \ C \ A \ T \ C \ A \ T \ G \ A \ T \ G \ A \ T \ C \ A \ T \ C$

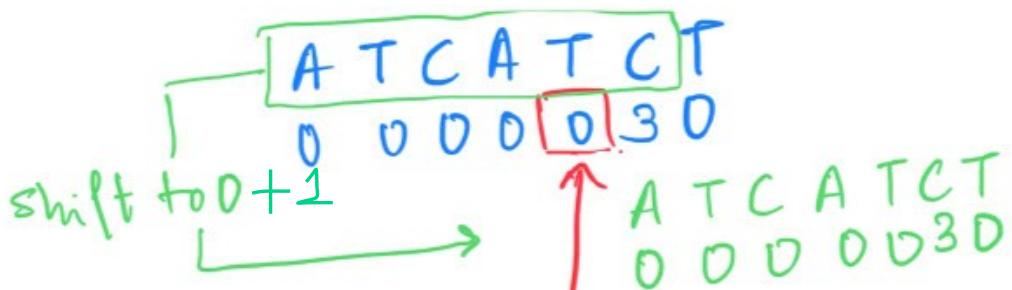


Because of this 2 we will shift pattern till 2+1 aligns with C

Step ① LPS'

C
↓

$S \rightarrow T \ C \ A \ T \ C \ A \ T \ G \ A \ T \ G \ A \ T \ C \ A \ T \ C$



Because of "0"
we were able to
shift directly to
start of pattern

We were able to
save one comparisi
n. This validates
the claim that
search process is
a bit more efficient

Q4 Solution

```
def z_arr_to_lps_prime(z_arr):
    lps_prime = [0] * len(z_arr)
    for i, z in enumerate(z_arr):
        if z > 0: lps_prime[i+z-1] = z

    return lps_prime

if __name__ == '__main__':
    print(z_arr_to_lps_prime(
        list(map(int, "0003000")))
    )
    print(z_arr_to_lps_prime(
        list(map(int, "001001001004000")))
    )
    print(z_arr_to_lps_prime(
        list(map(int, "00000207000050000")))
    ))
```

Example ①

T T T A 2 3 4 5 6 7

P[i] A T C A T C T

Z[i] 0 0 0 3 0 0 0

84.

lps[i] 0 0 0 1 2 3 0

lps' 0 0 0 0 0 3 0

②

P[i] A C A G T A T C A G T - A C A G

Z[i] 1 0 2 0 1 0 0 1 0 0 1 0 0 4 0 0 0

lps[i] 0 0 1 0 0 1 0 0 1 0 0 1 2 3 4

lps' 0 0 1 0 0 1 0 0 1 0 0 0 0 0 4

Example ③

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

ROHAN RO ROHAN RDHAN

Z 0 0 0 0 0 2 0 1 0 0 0 0 5 0 0 0 0 .

lps 0 0 0 0 0 1 2 1 2 3 4 5 6 7 3 4 5

lps' 0 0 0 0 0 0 2 0 0 0 0 0 0 7 0 0 5