

Raita Algoritması:

Raita Algoritması bir arama algoritmadır, özellikle satırın içinde kelimelerin aramak için kullanılır.

Aranacak kelimenin(pattern), son harften başlayarak aranacak metinden çıkarılan bir pattern ile karşılaştırılır, metinden çıkarılacak pattern aranacak kelimenin(pattern) karakter sayısına kadar aranacak metinden çıkarılacaktır.

Aranacak pattern: RAHIM

Text: ROBBIRAHIMKEREN

Text	R	O	B	B	I	R	A	H	I	M	K	E	R	E	N
Pattern	R	A	H	I	M										

İlk olarak RAHIM kelimesi ROBBİ kelimesiyle son harflarından başlayarak karşılaştırırız.

M ve I aynı olmadıkları için, sonraki aşamaya geçer:

Text	R	O	B	B	I	R	A	H	I	M	K	E	R	E	N
Pattern						R	A	H	I	M					

Bu aşamada M ve M aynı bulunur sonra arkaya giderek kalan karakterlerini karşılaştırır, bütün aranacak kelimenin karakterlerini aynı bulunca algoritma kelimeyi bulmuş olur.

Algoritma python dilinde:

```
# calculate bmbc
def Bmbc(pattern):
    bmbc_a = len(pattern) - 1
    bmbc = []
```

```

for i in range(len(pattern)):
    bmbc.append(bmbc_a - i)
return bmbc

def raita_algoritma(pattern, text):
    bmbc = Bmbc(pattern=pattern)
    found = 0

    # aranacak satırın harflerin sayısı pattern'in harfların sayısının
    # fazla harfleri silenecek
    while len(text) % len(bmbc) != 0:
        if text[len(text) - len(bmbc):] == pattern:
            found += 1
        text = text[:-1]

    control = False
    # Pattern aranacak satırın içinde araması
    for char_i in range(0, len(text), len(bmbc)):
        for i in range(len(bmbc)):
            if pattern[len(bmbc) - 1 - i] == text[(char_i + len(bmbc) - 1) -
i]:
                control = True
            else:
                control = False
                break
        if control == True:
            found += 1

    return int(found)

```

T(N):

Lt = len(text)

Lp = len(pattern)

$$T(N) = \sum_{n=0}^{Lt/Lp} \left[\sum_{m=0}^{Lp} 1 \right] + \sum_{i=0}^{Lp} 1 = \frac{Lt * Lp}{Lp} + Lp = Lt + Lp$$

Big O:

$$T(N) = \sum_{n=0}^{Lt/Lp} \left[\sum_{m=0}^{Lp} 1 \right] + \sum_{i=0}^{Lp} 1 = \frac{Lt * Lp}{Lp} + Lp = Lt + Lp$$

$$O(Lt + Lp) \Rightarrow T(N) \leq C * (Lt + Lp), N \geq N_0$$

Big Omega:

$$T(N) = \sum_{n=0}^{Lt/Lp} \left[\sum_{m=0}^{Lp} 1 \right] + \sum_{i=0}^{Lp} 1 = \frac{Lt * Lp}{Lp} + Lp = Lt + Lp$$

$$\Omega(Lt + Lp) \Rightarrow T(N) \geq C_1 * (Lt + Lp), N \geq N_0$$

Big Theta:

$$T(N) = \sum_{n=0}^{Lt/Lp} \left[\sum_{m=0}^{Lp} 1 \right] + \sum_{i=0}^{Lp} 1 = \frac{Lt * Lp}{Lp} + Lp = Lt + Lp$$

$$\Theta(Lt + Lp) \Rightarrow \Omega(Lt + Lp) \leq \Theta(Lt + Lp) \leq O(Lt + Lp)$$