Knuth-Morris-Pratt Algorithm (KMP)

8/6/20

KMP is a linear time string matching algorithm in O(n). KMP uses concept of prefix and suffix for generation of π-table.

Pattern a b c d a b c

Prefix LHS:

a, ab, abc, abcd ----

Suffix RHS

c, bc, abc, dabc

π-table i.e. longest prefix that same as suffix (LPS).

(1)

	1	2	3	4	5	6	7	8	9	10
	a	b	c	d	a	b	e	a	b	f
π-table	0	0	0	0	1	2	0	1	2	0

(2)

	1	2	3	4	5	6	7	8	9	10	11
	a	b	c	d	e	a	b	f	a	b	c
π-table	0	0	0	0	0	1	2	0	1	2	3

(3)

	1	2	3	4	5	6	7	8	9
	a	a	a	a	b	a	a	c	d
π-table	0	1	2	3	0	1	2	0	0

a a a a a a a

KMP

8 | 6 | 21

	1	2	3	4	5	6	7	8	9	10
Pattern	a	b	a	b	a	b	a	b	c	a
π-table	0	0	1	2	3	4	5	6	0	1

1 2 3
a b a b a
1 2 3

1 2 3 4
a b a b a b
1 2 3 4

1 2 3 4 5
a b a b a b a
1 2 3 4 5

1 2 3 4 5 6
a b a b a b a b
1 2 3 4 5 6

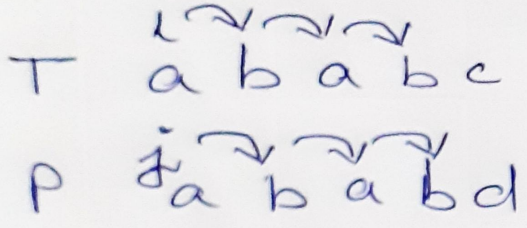
i=1 KMP (3)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T -	a	b	a	b	c	a	b	c	a	b	a	b	a	b	d

8 | 6 | 2 | 1

j=0

	1	2	3	4	5
P	a	b	a	b	d
π	0	0	1	2	0



j will be always one character back as we are comparing i and j+1

① Take two variable i and j

② Compare i and j+1, if match then move i and j to right

---

when i is at index 5 and j at index 4 i and j+1 mismatch then back track j by its π-value i.e. '2' here. So 'j' will be back at index '2'.

KMP

8 | 6 | 21

i

1	2	3	4	5	6	7	8	9	10	11	12
a	b	a	b	c	a	b	c	a	b	a	b	d

j 0

1	2	3	4	5
a	b	a	b	d
0	0	1	2	0

i will remain at index '5' but j will be back at index '2'

Now again start comparing i and j+1

i and j+1

c a mismatch

again j will backtrack at π-value '0', now j moves back at 'j=0'

again j+1 and i mismatch but now j can't move back from index '0' so now we have to move i.

Now i is at index '6' where 'a' is there and j+1 is also 'a'. so move right.

KMP

8/6/21

Again i at index 8 that is 'c' not matching j+1 in Pattern at index 3 i.e. a.

So backtrack 'j' by its π-value '0'

again j back at '0'.

again i at c not match with j+1 = i.e. 'a'

but j already at '0'

So it can't move back

So increment i to index 'a'

here it is 'a' that is matching with j+1 i.e. 'a'

---

again i at index 13 is 'a'

but j+1 at index '5' in pattern is 'd' so not match backtrack j at π-value of 'b' at ~~index~~ 2.

KMP

8/6/21

Again i at index 8 that is 'c' not matching j+1 in Pattern at index 3 i.e. a.

So backtrack 'j' by its π-value '0'

again j back at '0'.

again i at c still not match still j+1 = i.e. 'a'

but j already at '0' so it can't more back

So increment i to index 'a' here it is 'a' that is matching with j+1 i.e. 'a'

---

again i at index 13 is 'a'

but j+1 at index '5' in pattern is 'd' so ms match

backtrack j at π-value of 'b' at ~~index~~ 2.

KMP

8/6/21

i at index 13 is 'a'

j+1 at index 3 in pattern is 'a' match then move right both i and j