Shortest Way to form a String → Approach 1

Source = "a b c a" (0 1 2 3) target = "a a b a c b" (0 1 2 3 4 5)

result	sPos	tPos
1	0	0
1	1	1
1	2	1
1	3	1
1	4 → out of bounds (OOB)	2
∴ 2	0	2
2	1	2
2	2	3
2	3	3
2	4 → OOB	4
3	0	4
⋮ 3	2	4
3	3	5
3	4 → OOB	5
3	0	5
3	1	5
3	2	6 → OOB so stop

If any target char doesn't exist in source, return -1 early

↓

maintain hash set.

Shortest Way to form a String → Approach 2

Source = "a b c a" (0 1 2 3) target = "a a b a c b" (0 1 2 3 4 5)

map: (character, list of indices in source)

a → [0, 3] ⇒ list
b → [1]
c → [2]

sPos = 0
tPos = 0 char at sPos in source & tPos in target match, ∴ inc

⇒
sPos = 1
tPos = 1 char(tPos) = 'a'

∴ We're looking to see if sPos = 1 exists in list for 'a' in the map.
Since it doesn't exist, binary search returns: → = 1 → index in a's list where it should be present

-(insertionPoint) - 1
= -2

Make it positive ⇒ -(-2) - 1 = 2 - 1 = 1

∴ Next occurrence of a is at index 1 in the list.

∴ list[1] = 3

3 is the index in SPos where the next occurrence of a is found.

So, we set sPos to one index after 3.