LeetCode #242

Valid Anagram

Hash Table counting

The main idea for solving this problem is to count letters in both words and check whether the counts are equal. To reduce memory usage (though not asymptotically), we use a single hash table. First, we iterate through one of the strings and count the occurrences of each letters. Then, we iterate through the second string and check whether each character is present in the hash table. If a character is not found, we immediately return false. On each iteration, we decrement the count of the character in the hash table, and if the count reaches zero, we remove the key. After the loop finishes, we return true if the hash table is empty and false otherwise.

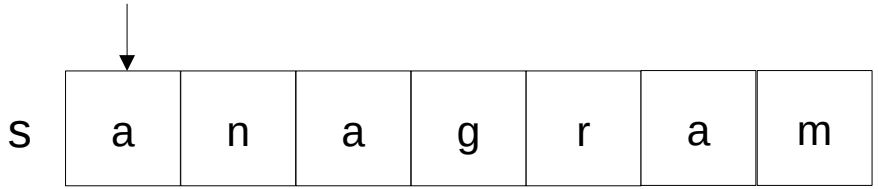
s	a	n	a	g	r	a	m

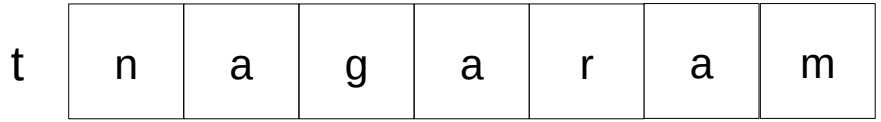
t	n	a	g	a	r	a	m

freq

key	value

The first step, we iterate through the sting s and count each character.

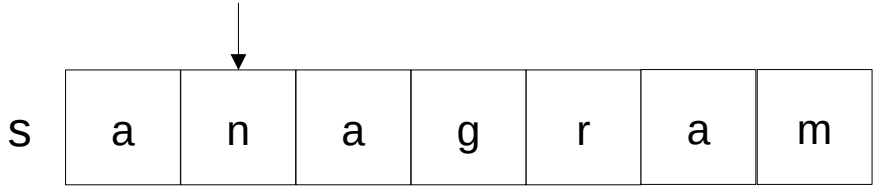


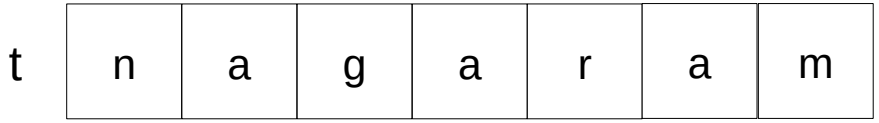


freq

key	value
a	1

freq[s[i]] += 1 = freq[‘a’] = 1

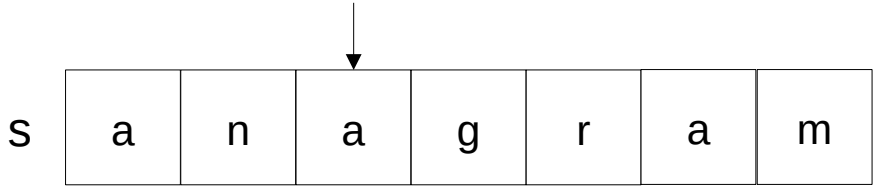


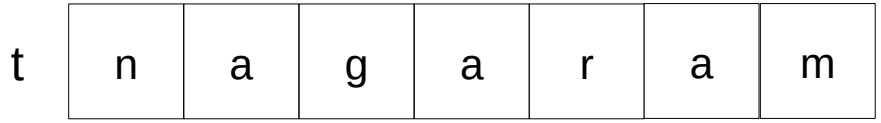


freq

key	value
a	1
n	1

freq[s[i]] += 1 = freq[‘n’] = 1

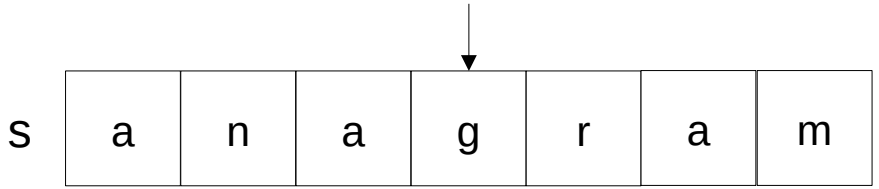


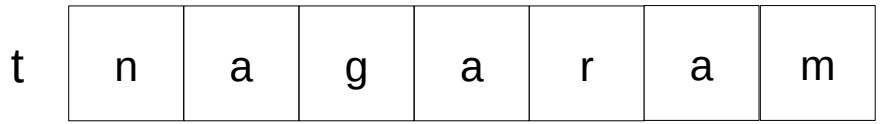


freq

key	value
a	2
n	1

freq[s[i]] += 1 = freq[‘a’] = 2

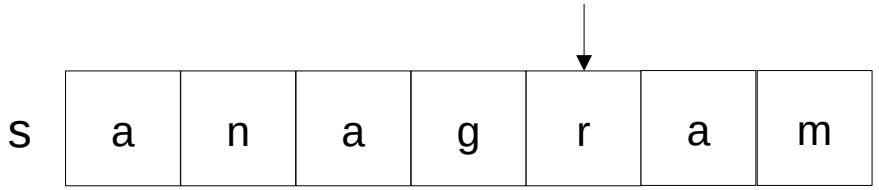


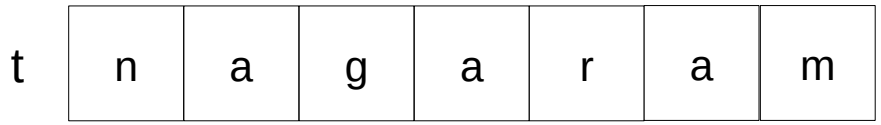


freq

key	value
a	2
n	1
g	1

freq[s[i]] += 1 = freq[‘g’] = 1

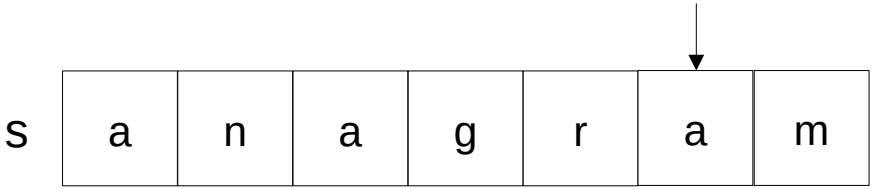


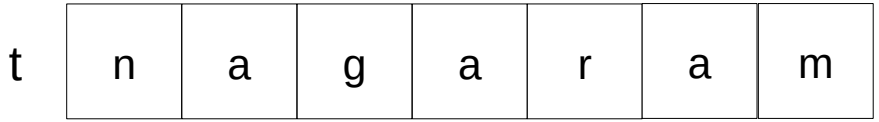


freq

key	value
a	2
n	1
g	1
r	1

freq[s[i]] += 1 = freq[‘r’] = 1

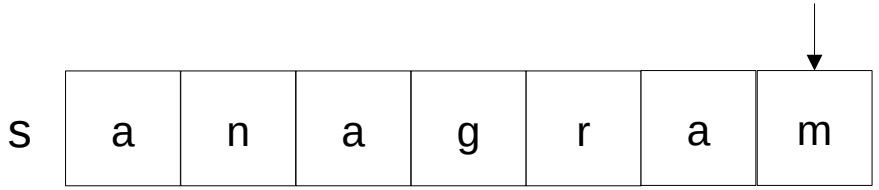


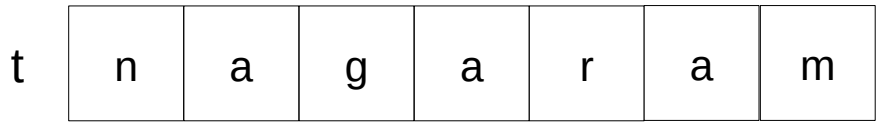


freq

key	value
a	3
n	1
g	1
r	1

freq[s[i]] += 1 = freq[‘a’] = 3





freq

key	value
a	3
n	1
g	1
r	1
m	1

freq[s[i]] += 1 = freq[‘m’] = 1

s

a	n	a	g	r	a	m

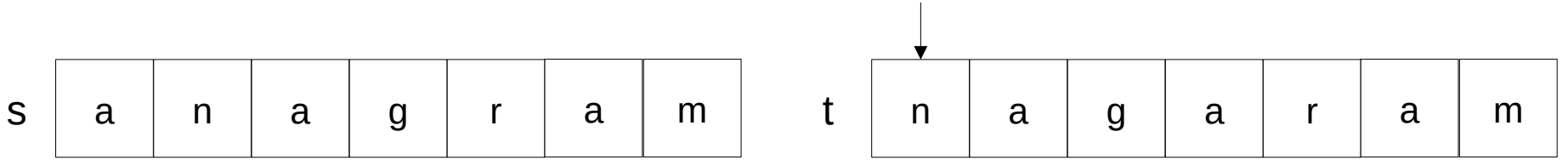
t

n	a	g	a	r	a	m

freq

key	value
a	3
n	1
g	1
r	1
m	1

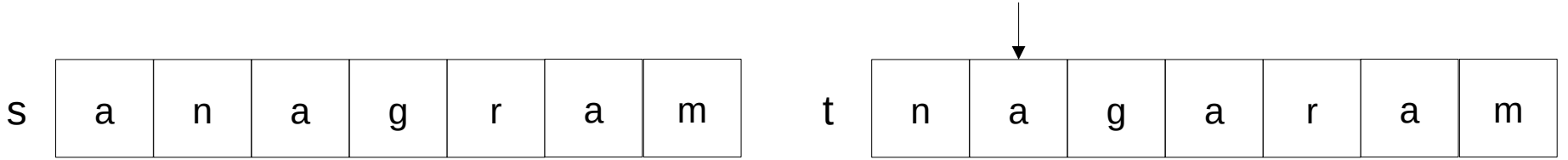
The second step, we iterate through the sting t and check whether a character is present in freq. If the character is not found, we return false, otherwise we decrement the count of the character in freq and continue. If the count reaches zero, we pop it.



freq

key	value
a	3
g	1
r	1
m	1

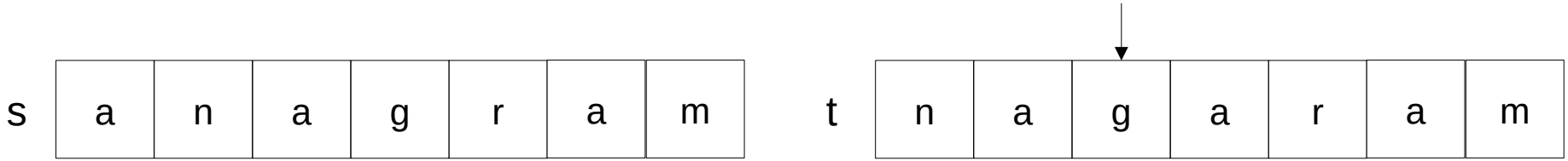
If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘n’] = 0 -> freq.remove(‘n’)



freq

key	value
a	2
g	1
r	1
m	1

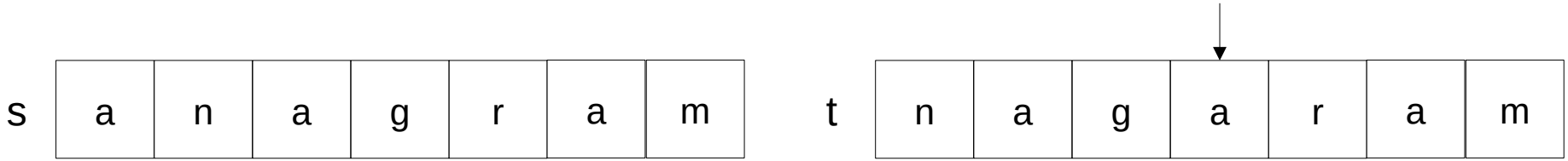
If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘a’] = 2



freq

key	value
a	2
r	1
m	1

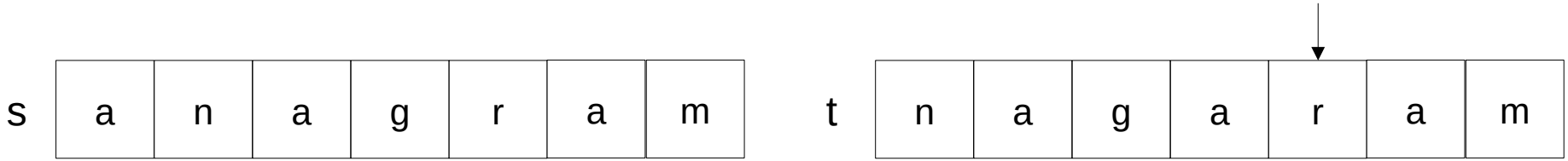
If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘g’] = 0 -> freq.remove(‘g’)



freq

key	value
a	1
r	1
m	1

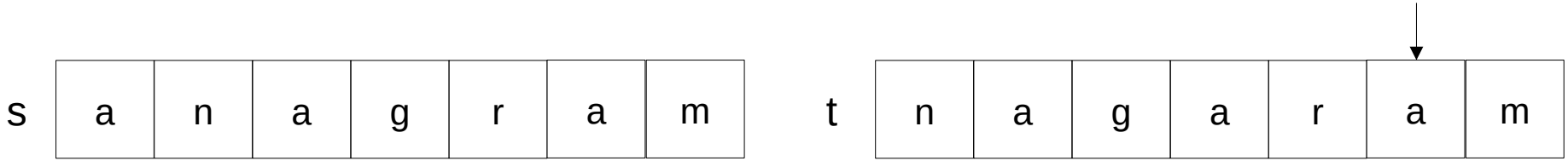
If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘a’] = 1



freq

key	value
a	1
m	1

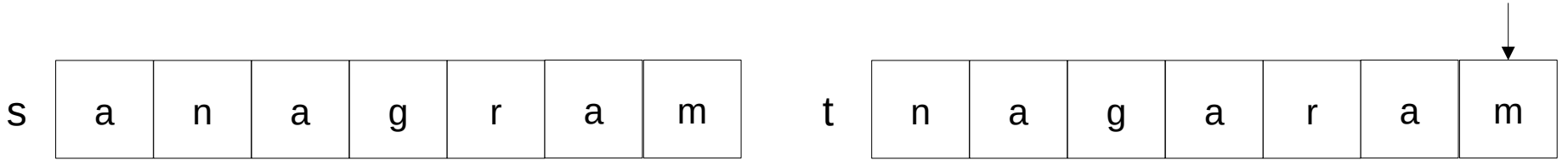
If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘r’] = 0 -> freq.remove(‘r’)



freq

key	value
m	1

If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘a’] = 0 -> freq.remove(‘a’)



freq

key	value

If t[i] not in freq: return false
freq[t[i]] -= 1 = freq[‘m’] = 0 -> freq.remove(‘m’)

s

a	n	a	g	r	a	m

t

n	a	g	a	r	a	m

freq

key	value

The final step, we check whether freq is empty. If it is, we return true, otherwise false.