

Valid anagram

part 1 using hash table

```
class Solution:
```

```
def valid_anagram(self, s: str, t: str) -> bool:
```

```
    if len(s) != len(t):  
        return False
```

```
    countS = {}
```

```
    countT = {}
```

```
    for i in range(len(s)):
```

```
        countS[s[i]] = 1 + countS.get(s[i], 0)
```

```
        countT[t[i]] = 1 + countT.get(t[i], 0)
```

```
    return countS == countT
```

```
print(Solution().valid_anagram(s="racecar", t="carrrce"))
```

Step 1:

self \rightarrow solution instance

countS = empty dictionary

countT = empty dictionary

Step 2 (for i in range(len(s)) :

$i \rightarrow 0$

Step 3 (count[s[i]] = 1 + countS.get(s[i], 0)
countS \rightarrow { "s": 1 }

Step 4 (count[t[i]] = 1 + countT.get(t[i], 0)
countT \rightarrow { "c": 1 }

\Rightarrow i is increasing by 1 for 7 times (len(s))
(0 to 6)

\Rightarrow the following steps are similar to Step 2, 3, 4
for exp:

Step 5 (for i in range(len(s)) :

$i \rightarrow 1$

Step 6 ($\text{countS}[s[i]] = 1 + \text{countS.get}(s[i], 0)$)
 $\text{countS} \rightarrow \{ "b": 1, "a": 1 \}$

Step 7 ($\text{countT}[t[i]] = 1 + \text{countT.get}(t[i], 0)$)
 $\text{countT} \rightarrow \{ "c": 1, "a": 1 \}$

... i increasing by 1 again

finally we get to have:

$i \rightarrow 6$ (the last digit)

$\text{countS} \rightarrow \{ "b": 2, "a": 2, "c": 2, "e": 1 \}$

$\text{countT} \rightarrow \{ "c": 2, "a": 2, "b": 2, "e": 1 \}$

we return True as they matched. ✓