

1. $O(n^2)$:

```
Function sum_equal_k_brute_force(array, n, k):  
    for i from 0 to n:  
        for j from i+1 to n:  
            if array[i] + array[j] = k:  
                return True  
    return False
```

2. $O(n)$:

```
Function sum_equal_k_counting_arr(array, n, k, maxValue=10000):  
  
    create countArray of size maxValue+1 and initialize to 0  
    for i from 0 to n:  
        countArray[array[i]] += 1  
  
    for j from 0 to n:  
        complement = k - array[j]  
        if complement >= 0 and complement < maxValue+1 and countArray[complement] != 0:  
            return True  
  
    return False
```

مقایسه حافظه مصرفی:

حافظه مصرفی در روش اول (بروت فورس) از $O(1)$ است. زیرا در این روش صرفاً به تعدادی متغیر برای اندیس‌ها تعریف می‌کنیم و در واقع به حافظه‌ای اضافی برای ورودی‌ها نیاز نداریم و مقایسه‌ها را مستقیماً روی آرایه ورودی انجام می‌دهیم. پس این روش از حافظه ثابت استفاده می‌کند.

حافظه مصرفی در روش دوم (آرایه شمارشی) از $O(\text{maxValue})$ است. در این روش به آرایه کمکی به طول maxValue نیاز داریم.

در نتیجه روش بروت فورس از لحاظ حافظه بیشتر کارآمد و ولی خوب از لحاظ زمان مصرفی شاید زیاد مناسب نباشد ولی روش آرایه شمارشی سریع‌تر و از لحاظ زمان مصرفی مناسب‌تر ولی خوب به حافظه مصرفی بیشتری نیاز دارد که می‌تونه با بزرگ بودن maxValue کمی مشکل ساز بشه.