

Coding Challenge

SumOfTwo

1. März 2020

1 Das Problem

Gegeben sind zwei `int`-Arrays `a` und `b`, mit beliebigen Werten und ein Zielwert (target value) `v`. Finde heraus, ob es ein Nummernpaar gibt, wobei ein Wert aus `a` und der andere Wert aus `b` sein muss, die in Summe den Zielwert `v` ergeben. Es soll `true` zurückgegeben werden, falls es so ein Paar gibt, andernfalls `false`.

2 Aufgabe

Schreibe die Methode `sumOfTwo(int[] a, int[] b, int v)` wie oben beschrieben. Die Lösung soll möglichst effizient sein, also **kein!!** Bruteforce.

3 Beispiele

Hier folgen zwei Beispiele für das oben beschriebene Problem:

```
int[] a = [1,2,3];  
int[] b = [10,20,30,40];  
int v = 42;
```

Die Methode `sumOfTwo` gibt `true` zurück, da $40 + 2 = 42$ ist.

a und b müssen nicht sortiert sein und können auch mehrmals den gleichen Wert beinhalten.

```
int[] a = {0,0,-5,30212};  
int[] b = {-10,40,-3,9};  
int v = -8;
```

Es wird wieder true returned, da $-5 + (-3) = -8$ ist.

4 Hinweise

- **Wichtig:** Denke daran, dass Arrays in Java $2^{31} - 1$ Elemente enthalten können → kein Bruteforce.
- HashSets sind hilfreich, da sie einfügen und lesen in effizienter und vor allem konstanter Zeit ermöglichen.
<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/HashSet.html>
- Natürlich könnt ihr die Aufgabe auch in Kotlin schreiben ☺

```
// HashSet für int - Werte  
HashSet<Integer> hashSet = new HashSet<>();
```

Mit Bruteforce ist gemeint, dass jedes einzelne Element aus a mit jedem einzelnen Element aus b addiert wird und danach mit v verglichen wird. Hier ein Beispiel:

$$\begin{aligned}a &= [1, 2, 3] \\ b &= [4, 5, 6] \\ v &= 7\end{aligned}$$

$$\begin{aligned}a[0] &= 1 \\ b[0] + a[0] &= 7? \\ &\cdot \\ &\cdot \\ b[3] + a[0] &= 7 \\ &\text{return true}\end{aligned}$$