

Aufgabe 41)

a)

Anmerkungen zur folgenden Lösung:

- Unter geordneten Paaren ist zu verstehen, das die Paare im Ausgabearray geordnet sind, also:

$\text{Ausgabe}[i][k] < \text{Ausgabe}[j][k]$

$\text{Ausgabe}[k][i] < \text{Ausgabe}[k][j]$

Gibt es zwei Elemente a, b deren Summe den geforderten Wert ergibt, resultieren aus den beiden Elementen auch zwei Paare. Nämlich das Paar (a,b) und das Paar (b,a). Hierbei handelt es sich nicht um das gleiche Paar. (Vergleichbar mit gerichteten Kanten in einem Graphen: Kante von a nach b \neq Kante von b nach a)

- In der Lösung wurde Mergesort zum Sortieren des Array verwendet, um hinterher eine binäre Suche durchführen zu können. Der Mergesort Algorithmus arbeitet jedoch nicht in Place, möchte man in Place Sortieren wäre die Verwendung des Heapsort (auch $O(n \log n)$) möglich.

Es sind auch andere Lösungen denkbar. Zum Beispiel einfügen der Elemente in einen AVL-Baum und dann die Suche innerhalb dieses Baumes. Allerdings arbeitet auch dieser Algorithmus nicht in Place (aber in $O(n \log n)$)

- Der Algorithmus zur Suche der Paare wird zweimal durchlaufen, da die Größe eines Array bei der Erzeugung festgelegt werden muss. Eine andere Möglichkeit wäre es die Paare zunächst in einem anderen Datentyp (z.B. Vector) zu speichern, der dynamisch erweitert werden kann und diesen dann in `int[][]` umzuwandeln.

Umgangssprachlich:

- 1.) Zunächst wird das Array mit einem Sortierv erfahren das in $O(n \log n)$ Zeit sortiert.
- 2.) Jetzt muss zunächst die Größe des 2-Dimensionalen-Ergebnisarrays berechnet werden. Dazu wird für jedes Element des Array, (Summe – Element) gerechnet und mit Hilfe einer binären Suche in $O(\log n)$ Zeit überprüft, ob sich die resultierende Zahl ebenfalls im Array befindet. Ist dies der Fall wird eine Zählvariable erhöht. Wenn alle Elemente durchlaufen wurden enthält diese Variable die Anzahl der Paare.
- 3.) Nun wird das Ergebnisarray in passender Größe erzeugt und die im Schritt 2 beschriebene Suche nach den Paaren erneut durchgeführt. Wird eines gefunden trägt man es in das Ergebnisarray ein.
- 4.) Das Ergebnisarray wird zurückgegeben

In Pseudocode:

```
public int[][] computePairsForSum(int[] values, int sum){  
    sortiere das Array in O(n log n) Zeit (z.B. Mergesort);  
    zähle Paare und erzeuge passendes Array;  
    for (jedes Element){  
        if(ist Summe-Element im Array enthalten (binäre Suche)){  
            trage (Element, Summe-Element) ins Ergebnisarray ein;  
        }  
    }  
    return Ergebnisarray;  
}
```

