Felix Janssen
Benedikt Rips
Marcel Schoppmeier

Aufgabe 29

(a) Algorithmus zum Auffinden geordneter Paare

Vorraussetzungen Sei $M := \{a_1, \ldots, a_n\}$, $a_i \in \mathbb{Z}$ mit $a_i \neq a_j$ für $i \neq j$ eine Menge paarweise verschiedenener ganzer Zahlen. Zu einer gegebenen ganzen Zahl $s \in \mathbb{Z}$ werden nun alle geordneten Paare $(x, y) \in M \times M$ gesucht mit $x + y = s$.

Funktionsweise Zunächst einmal wird die Menge M mittels Heap-Sort sortiert. Dadurch erhalten wir eine sortierte Zahlenfolge:

$$(a_{i_1}, \ldots, a_{i_n}) \text{ mit } a_{i_l} \in M \; \forall 1 \leq l \leq n, \qquad i \in \{1 \ldots n\} \text{ mit } i_k \neq i_l \text{ für } k \neq l$$

D.h. es gilt für alle $1 \leq k < l \leq n$: $\quad a_{i_k} < a_{i_l}$

(Es gilt *echt* kleiner, da alle Zahlen paarweise verschieden)

Jetzt ist es ziemlich einfach, die geordneten Paare mit der gewünschten Eigenschaft zu finden. Dafür: Zu jedem a_{i_k} sei $c_{i_k} := s - a_{i_k}$ das Komplement. Nun wird überprüft, ob das Komplement c_{i_k} in $(a_{i_{k+1}}, \ldots, a_{i_n})$ enthalten ist. Wenn dies der Fall ist, werden die geordneten Paare (a_{i_k}, c_{i_k}) und (c_{i_k}, a_{i_k}) der Ergebnismenge E hinzugefügt. Nachdem dieser Vorgang für alle $1 \leq k < n$ durchgeführt wurde, sind alle geordneten Paare bestimmt. Danach wird die Ergebnismenge E zurückgegeben und der Algorithmus terminiert.

Laufzeit Heap-Sort benötigt, wie bekannt, eine Laufzeit von $\mathcal{O}(n \log n)$. Die Überprüfung, ob c_i enthalten ist, erfolgt mit binärem Suchen, dementsprechend benötigt dieser Schritt eine Laufzeit von $\mathcal{O}(\log n)$. Insgesamt ergibt sich dann die Laufzeit $T(n)$ für beide Schritte als:

$$T(n) = f + g \qquad \text{mit } f \in \mathcal{O}(n \log n),\, g \in \mathcal{O}(\log n)$$

Also gilt für die Laufzeit dieses Algorithmus: $T(n) \in \mathcal{O}(n \log n)$.