

Erweitern Sie Ihre Implementierung **ADS_set** um die Methode

bool y(const key_type& a, const key_type& b) const;

Diese soll genau dann **true** liefern, wenn in der durch einen Iterator gelieferten Reihenfolge der Schlüsselwerte

- **a** unmittelbar hinter **b**, oder
- **b** unmittelbar hinter **a** liegt.

Erlaubt ist nur das Verwenden von **std::equal_to** für den Vergleich von Schlüsselwerten. Aufruf von anderen Methoden oder Funktionen, insbesondere die Verwendung von Iteratoren (und damit auch die Verwendung einer range based for loop), ist nicht erlaubt.

Die Zeitkomplexität der Funktion **y** muss $O(n)$ sein (n ist dabei die Anzahl der Elemente im Set), die Speicherkomplexität $O(1)$. Es ist beispielsweise nicht erlaubt, zusätzliche Felder mit einer nicht konstanten Größe zu verwenden.

Beispiele:

Der Iterator liefert alle gespeicherten Schlüsselwerte in der Reihenfolge (4,7,1,5,3,6,0,8,10,2,9)

- **y(5,6)** und **y(7,7)** liefern **false**
- **y(5,3)** und **y(10,8)** liefern **true**
- **y(1,12)** und **y(20,12)** liefern **false**

Eine mathematische Formulierung der Aufgabenstellung für jene, die diese Art der Beschreibung bevorzugen: Sei $A(a_i)$ die Folge der Schlüsselwerte, wie sie vom Iterator geliefert werden, dann liefert

$$y(x,y): \begin{cases} true, & \text{wenn } x = a_i \wedge y = a_j \wedge |i - j| = 1 \\ false & \text{sonst} \end{cases}$$

Tipp:

Eine einfache Methode ist, die Werte in Iteratorreihenfolge zu durchlaufen (allerdings, ohne Iteratoren zu verwenden) und dabei zu überprüfen, ob die beiden als Parameter empfangenen Schlüsselwerte unmittelbar hintereinander auftreten.