

Der bestehende Iterator Ihres **ADS_set** ist um einen zusätzlichen „Modus“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle Elemente in einer beliebigen Reihenfolge, wobei die Reihenfolge immer dieselbe sein muss, solange das **ADS_set** nicht geändert wird (Modus „normal“). Im neuen Modus „speziell“ sollen die Werte in derselben Reihenfolge geliefert werden, aber es werden alle Werte vor einem bestimmten Wert übersprungen, ebenso werden alle Werte ab einem bestimmten Wert übersprungen. In beiden Modi erreicht der Iterator **end()**, sobald keine Elemente mehr vorhanden sind.

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

const_iterator x(const key_type &a, const key_type &b) const;

Diese soll einen Iterator im Modus „speziell“ liefern. Wenn **a** oder **b** nicht im **ADS_set** enthalten sind, oder wenn **a** in Iteratorreihenfolge nicht vor **b** liegt, dann gilt **x() == end()**. Andernfalls liefert **x()** einen Iterator, der auf **a** zeigt. Der von **x()** gelieferte Iterator soll beim Inkrementieren in weiterer Folge alle Werte in Iteratorreihenfolge liefern, allerdings nur bis (exklusive) **b**. Sobald **b** erreicht wird, wird **end()** geliefert.

Die Zeitkomplexität und Speicherkomplexität der Operatorfunktionen müssen unverändert bleiben. So sind zB zusätzliche Felder mit nicht konstanter Größe unzulässig.

Der Vergleich von zwei Werten **u, v** vom Typ **T** erfolgt mittels **std::equal_to<T>{}(u, v)**.

Beispiele: Angenommen der von **begin()** retournierte Iterator liefert alle gespeicherten Schlüsselwerte in der Reihenfolge (4,7,1,5,3,6,0,8,10,2,9)

dann liefert der Aufruf von	den Iterator	und wenn dieser Iterator bis end() inkrementiert wird, liefert er die folgenden Schlüsselwerte in der Reihenfolge
x(4,9)	Iterator auf 4	(4,7,1,5,3,6,0,8,10,2)
x(5,0)	Iterator auf 5	(5,3,6)
x(1,1)	end()	-
x(4,666)	end()	-
x(666,4)	end()	-
x(6,3)	end()	-
x(666,666)	end()	-

Anleitung: Schreiben Sie **keine** neue Iteratorklasse! Erweitern Sie die bestehende Iterator-Klasse wie folgt:

- Es ist zu speichern, ob der Iterator im Modus „normal“ oder „speziell“ ist und die für den Modus „speziell“ erforderlichen zusätzlichen Informationen.
- Es muss ein Iterator im Modus „speziell“ erzeugt werden können. Dazu ist ein neuer Konstruktor zu schreiben (und/oder bestehende zu erweitern) um die Instanzvariablen entsprechend zu initialisieren.

Die oben beschriebene Methode **ADS_set::x** erzeugt einen Iterator im Modus „speziell“ und retourniert diesen.

Die Methode **ADS_set::begin** liefert wie bisher einen Iterator im Modus „normal“.