

Der bestehende Iterator Ihres **ADS_set** ist um einen zusätzlichen „Modus“ zu erweitern. In der bisherigen Implementierung liefert der Iterator alle n Elemente des **ADS_set** in einer beliebigen Reihenfolge, wobei die Reihenfolge immer dieselbe sein muss, solange das **ADS_set** nicht geändert wird (Modus „normal“). Im neuen Modus „speziell“ liefert der Iterator die Elemente in derselben Reihenfolge, es werden aber nach dem ersten Element nur mehr genau jene Elemente geliefert, die größer¹ als das erste gelieferte Element sind. Alle anderen Elemente werden übersprungen. In beiden Modi erreicht der Iterator nach dem letzten gelieferten Element **end()**.

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

```
const_iterator z() const;
```

z() erzeugt einen Iterator im Modus „speziell“. Wenn kein Element im **ADS_set** vorhanden ist, dann gilt **z() == end()**.

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

Beispiele:

Angenommen der von begin() retournierte Iterator liefert alle n gespeicherten Elemente in der Reihenfolge	Dann liefert der von z() retournierte Iterator die folgenden Elemente in der folgenden Reihenfolge
(1,2,4,3,5,6,7)	(1,2,4,3,5,6,7)
(4,2,3,1,5,6)	(4,5,6)
(4,2,6,5,1)	(4,6,5)
(5,3,4,1,2)	(5)
(7,8)	(7,8)
(7,6)	(7)
(7)	(7)
()	()

Anleitung: Schreiben Sie **keine** neue Iteratorklasse! Erweitern Sie die bestehende Iterator-Klasse wie folgt (dies ist nur einer der möglichen Lösungsansätze, abweichende korrekte Lösungen sind natürlich zulässig):

- Es muss ein Iterator im Modus „speziell“ erzeugt werden können. Dazu ist ein neuer Konstruktor zu schreiben und/oder bestehende zu erweitern. Eventuell benötigen Sie zusätzliche Konstruktorparameter und/oder Instanzvariablen.
- Passen Sie die Inkrement-Operationen (nur!) für den Modus „speziell“ an: Alle Elemente, die nicht größer sind als das erste gelieferte Element, werden übersprungen. Nach dem letzten zu liefernden Element wird der Iterator (wie immer) auf **end()** gesetzt.

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

¹ Der Aufruf **std::less<value_type>{}(value1, value2)** liefert **true**, falls **value1** kleiner als **value2** ist, und **false** sonst