

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

size_t x() const;

Diese soll den Abstand zwischen dem kleinsten und dem zweitkleinsten gespeicherten Wert in Bezug auf die von einem Iterator gelieferten Reihenfolge der Elemente retournieren (als Absolutbetrag, also nicht negativ). Wenn weniger als zwei Werte im **ADS_set** enthalten sind, dann liefert **x** eine Exception vom Typ **std::runtime_error** (gegebenenfalls müssen Sie zusätzlich **<stdexcept>** inkludieren).

Für den Vergleich von Werten ist **std::less** (bzw. der alias **key_compare**, sofern vorhanden) zu verwenden. Der Aufruf **std::less<T>{}(x,y)** für die beiden Werte **x** und **y** vom Typ **T** liefert **true**, falls **x** kleiner als **y** ist, und **false** sonst.. Ein 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 **x** 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:

Angenommen der von begin() retournierte Iterator liefert alle gespeicherten Elemente in der Reihenfolge	dann liefert x()	denn das kleinste und das zweitkleinste Element sind
(7,4,1,5,3,6,0,8,10,2,9)	4	0, 1
(1,7,8,4,5)	3	1, 4
(7,8,9)	1	7, 8
(9,7)	1	7, 9
(7)	std::runtime_error	n.a.
()	std::runtime_error	n.a.

Anleitung:

Eine einfache Methode ist, die Werte in Iteratorreihenfolge zu durchlaufen (allerdings, ohne Iteratoren zu verwenden) und dabei den kleinsten und den zweitkleinsten Wert zu ermitteln, wobei auch die jeweilige „Position“ (in Bezug auf die Iteratorreihenfolge) der gefundenen Werte zu speichern ist. Die Differenz (Absolutbetrag) der beiden Positionen ist das Ergebnis von **x()**.

Sind weniger als zwei Werte vorhanden, ist eine Exception vom Typ **std::runtime_error** zu werfen.