

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

```
size_t x(const key_type& a, const key_type& b) const;
```

Diese soll den Abstand der Werte **a** und **b** in Bezug auf die von einem Iterator gelieferten Reihenfolge der Schlüsselwerte retournieren (als Absolutbetrag, also nicht negativ). Wenn zumindest einer der beiden Werte nicht im **ADS_set** enthalten ist, dann liefert **x** eine Exception vom Typ **std::runtime_error** (gegebenenfalls müssen Sie zusätzlich **<stdexcept>** inkludieren).

Erlaubt ist nur das Verwenden von **std::equal_to** (bzw. der alias **key_equal**) für den Vergleich von Schlüsselwerten. 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 Iterator Ihres **ADS_set** liefert alle gespeicherten Schlüsselwerte in der Reihenfolge (4,7,1,5,3,6,0,8,10,2,9)

- **x(5,0)** und **x(0,5)** liefern jeweils **3** (0 befindet sich 3 Positionen nach 5)
- **x(5,3)** und **x(3,5)** liefern jeweils **1** (3 befindet sich 1 Position nach 5)
- **x(5,5)** liefert **0** (5 befindet sich an derselben Position wie 5)
- **x(1,12)** und **x(20,12)** liefern jeweils eine Exception **std::runtime_error** (12 und 20 sind nicht vorhanden)

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

$$x(a,b) := \begin{cases} |i-j|, & \text{wenn } a = s_i \wedge b = s_j \\ \text{std} :: \text{runtime_error}, & \text{sonst} \end{cases}$$

Tipp:

Eine einfache Methode ist, die Werte in Iteratorreihenfolge zu durchlaufen (allerdings, ohne Iteratoren zu verwenden). Sobald der erste Wert (**a** oder **b**) erreicht ist, sind die Schritte zu zählen, bis der andere Wert erreicht ist. Wird nur einer der beiden Werte gefunden, oder gar keiner, ist eine Exception vom Typ **std::runtime_error** zu werfen.