

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

```
statistics_t z() const;
```

Diese soll für die drei **insert**-Methoden die Gesamtanzahl der jeweils *eingefügten* bzw. *nicht eingefügten* Werte liefern („Statistikdaten“). Ein „nicht eingefügter Wert“ ist ein Wert, der an eine **insert**-Methode übergeben wurde, aber zu diesem Zeitpunkt schon im **ADS_set** vorhanden ist. **statistics_t** ist bereits definiert und ist wie folgt aufgebaut:

```
struct statistics_t {  
    size_t ok[3];    // eingefügte Werte  
    size_t fail[3]; // nicht eingefügte Werte (weil schon vorhanden)  
};
```

Der Defaultkonstruktor von **statistics_t** initialisiert alle Werte mit 0, Kopierkonstruktor, Kopierzuweisungsoperator, Ausgabeoperator, **swap** und die Vergleichsoperatoren (**==**, **!=**) funktionieren wie erwartet. Die Zuordnung der Indexpositionen in **ok** und **fail** zu den drei **insert**-Methoden ist wie folgt:

Index 0: **void insert(std::initializer_list<key_type>);**

Index 1: **std::pair<iterator,bool> insert(const key_type &);**

Index 2: **template<typename InputIt> void insert(InputIt, InputIt);**

Zu diesem Zweck sind die Statistikdaten im **ADS_set** zu verwalten und bei den drei **insert**-Methoden entsprechend anzupassen. Achtung: es sind nur Einfügeversuche im Zuge von „externen“ Aufrufen der **insert**-Methoden zu berücksichtigen. Externe Aufrufe sind Aufrufe von außerhalb des **ADS_set** (also hier vom Unit-Test), nicht hingegen Aufrufe der **insert**-Methoden durch andere Methoden des **ADS_set** („interne“ Aufrufe).

Beim Kopierkonstruktor und Kopierzuweisungsoperator

```
ADS_set(const ADS_set &other)  
ADS_set &operator=(const ADS_set &other);
```

werden die Statistikdaten aus **other** übernommen.

Bei allen anderen Konstruktoren sowie beim Initializer-List-Zuweisungsoperator

```
ADS_set &operator=(std::initializer_list<key_type> ilist);
```

werden die Statistikdaten in den Anfangszustand (alles 0) gesetzt.

Bei **swap**-Operationen werden die Statistikdaten der beiden **ADS_sets** getauscht. Bei **clear** und **erase** bleiben die Statistikdaten unverändert. Bei den Vergleichsoperatoren (**==**, **!=**) werden die Statistikdaten ignoriert.

Die Zeitkomplexität von **z()** muss $O(1)$ sein, die Speicherkomplexität $O(1)$. Die Zeit- und Speicherkomplexität aller übrigen Methoden und Funktionen müssen unverändert (spezifikationskonform) bleiben. Für die Verwendung der STL gelten dieselben Regeln wie im übrigen Projekt.

Hinweis: Falls eine **insert**-Methode sowohl extern als auch intern aufgerufen wird, kann man die Unterscheidung zB durch Hinzufügen eines weiteren Parameters realisieren. Dieser Parameter erhält einen Vorgabewert (für externe Aufrufe) und einen anderen Wert bei internen Aufrufen.