

# 11 Collection

Last updated by | kees.vanloenen | Jun 29, 2025 at 9:38 AM GMT+2

## 11 Collection

### Zwarte Piste (Ri)

#### 0 Test-driven

Doe onderstaande oefeningen "test-driven"

### Blauwe Piste (Shu)

#### 1 Maak een class `sortedIntList`

Zoals hieronder beschreven, maar dan zonder dat hij gesorteerd hoeft te zijn.

### Rode Piste (Ha)

#### 2 Maak een class `SortedList`

- Een `SortedList` heeft een **constructor**

```
SortedList sorted = new SortedList();
```



- Een `SortedList` heeft een `Add(int item)` methode. Deze methode voegt de item toe aan de lijst en zorgt er voor dat de items in de lijst altijd in oplopende volgorde staan.

```
sorted.Add(5);  
sorted.Add(2);  
sorted.Add(7);  
sorted.Add(3);
```



```
// sorted is nu een lijst 2-3-5-7
```

- Een `SortedList` heeft een **readonly indexer**:

```
int secondElement = sorted[1]; // = 3
```



```
sorted[1] = 5; // does not compile
```



- Een `SortedList` kan **flexibel groeien**. Initieel heeft de `SortedList` een capaciteit van 4 (de interne `int[]` heeft een `Length` van 4). Maar als we meer elementen toevoegen, groeit het array dynamisch mee.
- In de constructor van de `SortedList` kun je **optioneel een andere capaciteit** aangeven.

```
SortedIntList groteLijst = new SortedIntList(100);  
// een sortedlist met een interne int[] met Length van 100.
```



- Een sortedIntList implementeert **IEnumerable**

```
SortedIntList primes = { 11,3,5,2,7 };  
foreach (int p in primes)  
{  
    Console.Write(p + "-");  
}  
  
// produces 2-3-5-7-11
```

