

```
r
}
```

Saat käyttää `scala.collections.immutable`-kirjaston luokkia ja metodeja.

Vihje: ainakin `filter` ja `sum` saattavat olla käteviä metodeja.

6. Anna kummallekin alla olevalle funktiolle ajoaikaaraja suhteessa argumenttisekvenssin s pituuteen (merkitse sitä muuttujalla n) käyttäen \mathcal{O} -notaatiota. Antamiesi ajoaikaarajafunktioiden tulee olla mahdollisimman hitaasti kasvavia.

Muistathan, että `Array`-tyypillä indeksoitu elementin hakeminen ja `length`-metodi ovat vakioaikaisia operaatioita.

Function 1:

```
def myFunc1(s: Array[Int]): Int = {
  require(s.nonEmpty)

  def myMax(): Int = {
    var r = s(0)
    var i = 1
    while(i < s.length) {
      if(s(i) > r)
        r = s(i)
      i = i + 1
    }
    r
  }

  var i = 0
  var freq = 0
  while(i < s.length) {
    if(s(i) == myMax())
      freq = freq + 1
    i = i + 1
  }
  freq
}
```

$\mathcal{O}(n^2)$

$n \cdot n$

Function 2:

```
def myFunc2(s: Array[Int]): Int = {
  require(s.nonEmpty)

  def myMax(): Int = {
    var r = s(0)
    var i = 1
    while(i < s.length) {
      if(s(i) > r)
        r = s(i)
      i = i + 1
    }
    r
  }

  var i = 0
  var freq = 0
```