

# TÖL212M Röktudd Forritun - Einstaklingsverkefni 3

Andri Fannar Kristjánsson

31. janúar 2025

## Einstaklingsverkefni 3

### 1

Klárið að forrita Dafny skrána E3-skeleton.dfy

#### 1.1 Svar:

Hér fyrir neðan má sjá leystu útgáfuna sem Dafny samþykkir. Hægt er einnig að sjá kóðann hér: <https://shorturl.at/liWI2>.

```
// Höfundur spurningar: Snorri Agnarsson, snorri@hi.is

// Höfundur lausnar: Andri Fannar Kristjánsson, afk6@hi.is
// Permalink lausnar: https://shorturl.at/liWI2

////////////////////////////////////
// Hér byrjar óbreytanlegi hluti skrárinnar.
// Fyrir aftan þann hluta er sá hluti sem þið eigið að breyta.
//
// This is the start of the part of the file that should not
// be changed. Following this part is the part you should
// modify.
////////////////////////////////////

// Hjálparfall sem finnur minnsta gildi í poka.
// Helper function that finds the smallest value in a multiset.
method MinOfMultiset( m: multiset<int> ) returns( min: int )
  requires m != multiset{}
  ensures min in m
  ensures forall z | z in m :: min <= z
{
  min :| min in m;
  var done := multiset{min};
  var m' := m-done;
  while m' != multiset{}
  {
    decreases m'
    invariant m == done+m'
    invariant min in done
    invariant forall z | z in done :: min <= z
    {
      var z :| z in m';
      done := done+multiset{z};
      m' := m'-multiset{z};
      if z < min { min := z; }
    }
  }
}
```

```

// Ekki má breyta þessu falli.
// Do not change this function.
method Test( m: multiset<int> )
{
    var s := Sort(m);
    assert multiset(s) == m;
    assert forall p,q | 0 <= p < q < |s| :: s[p] <= s[q];
}

method Main()
{
    var m := multiset{0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9};
    var s := Sort(m);
    assert multiset(s) == m;
    assert forall p,q | 0 <= p < q < |s| :: s[p] <= s[q];
    print s;
}

////////////////////////////////////
// Hér lýkur óbreytanlega hluta skrárinnar.
// Hér fyrir aftan er sá hluti sem þið eigið að breyta til að
// útfæra afbrigði af selection sort.
//
// This is the end of the part of the file that should not be
// changed. The subsequent part is the part you should change
// in order to implement a version of selection sort.
////////////////////////////////////

// Selection sort sem raðar poka í runu.
// Klárið að forrita þetta fall.
// Selection sort that sorts a multiset into a sequence.
// Finish programming this function.
method Sort( m: multiset<int> ) returns ( s: seq<int> )
    // Setjið viðeigandi ensures klausur hér
    // Put appropriate ensures clauses here
    ensures forall p,q | 0 <= p < q < |s| :: s[p] <= s[q]
    ensures multiset(s) == m
{
    // Setjið viðeigandi frumstillingar á m' og s hér.
    // m' er ný staðvær breyta en s er skilabreyta.
    // Put appropriate initializations for m' and s here.
    // m' is a new variable but s is the return variable.
    s := [];
    var m' := m;
    while m' != multiset{}
        // Ekki breyta fastayrðingu lykkju
        // Do not change the loop invariant
        decreases m'
        invariant m == m'+multiset(s)
        invariant forall p,q | 0 <= p < q < |s| :: s[p] <= s[q]
        invariant forall z | z in m' :: forall r | 0 <= r < |s| :: z >= s[r]
    {
        // Setjið viðeigandi stofn í lykkjuna hér
        // Put an appropriate body of the loop here

        // Find the minimum value in the multiset,
        // remove it and add it to the sequence.
        var x := MinOfMultiset(m');

```

```
    m' := m' - multiset{x};  
    s := s + [x];  
  }  
  
  return;  
}
```