

```

// Höfundur spurningar: Snorri Agnarsson, snorri@hi.is
// Permalink spurningar: https://rise4fun.com/Dafny/G4sc3

// Höfundur lausnar: Alexander Guðmundsson
// Permalink lausnar: https://rise4fun.com/Dafny/nujsu

// Insertion sort með hjálp helmingunarleitar.

method Search( s: seq<int>, x: int ) returns ( k: int )
  // Ekki má breyta forskilyrðum eða eftirskilyrðum fallsins
  requires forall p,q | 0 <= p < q < |s| :: s[p] <= s[q];
  ensures 0 <= k <= |s|;
  ensures forall i | 0 <= i < k :: s[i] <= x;
  ensures forall i | k <= i < |s| :: s[i] >= x;
  ensures forall z | z in s[..k] :: z <= x;
  ensures forall z | z in s[k..] :: z >= x;
  ensures s == s[..k]+s[k..];
{
  // Setjið viðeigandi stofn fallsins hér.
  var p := 0;
  var q := |s|;

  if p == q
  {
    return p;
  }
  while p != q
  {
    decreases q-p;
    invariant 0 <= p <= q <= |s|;
    invariant forall r | 0 <= r < p :: s[r] <= x;
    invariant forall r | q <= r < |s| :: s[r] >= x;

    {
      var m := p + (q-p)/2;
      if s[m] == x
      {
        return m;
      }
      if s[m] < x
      {

```

```

        p := m+1;
    }
    else
    {
        q := m;
    }
}

return p;

}

method Sort( m: multiset<int> ) returns ( r: seq<int> )
    ensures multiset(r) == m;
    ensures forall p,q | 0 <= p < q < |r| :: r[p] <= r[q];
{
    // Setjið viðeigandi frumstillingu á r og rest hér.
    // r er skilabreyta en rest er ný breyta sem þið búið til.
    r := [];
    var rest := m;
    while rest != multiset{}
        // Ekki breyta fastayrðingu lykkju
        decreases rest;
        invariant m == multiset(r)+rest;
        invariant forall p,q | 0 <= p < q < |r| :: r[p] <= r[q];
    {
        // Setjið viðeigandi stofn lykkjunnar hér.
        // Fjarlægjið eitt gildi úr rest með
        //     var x :| x in rest;
        //     rest := rest-multiset{x};
        // og notið Search til að finna réttan stað
        // í r til að stinga [x] inn í r.
        var x :| x in rest;
        rest := rest - multiset{x};
        var k := Search(r, x);
        r := r[..k] + [x] + r[k..];
    }
    return r;
}

```

}