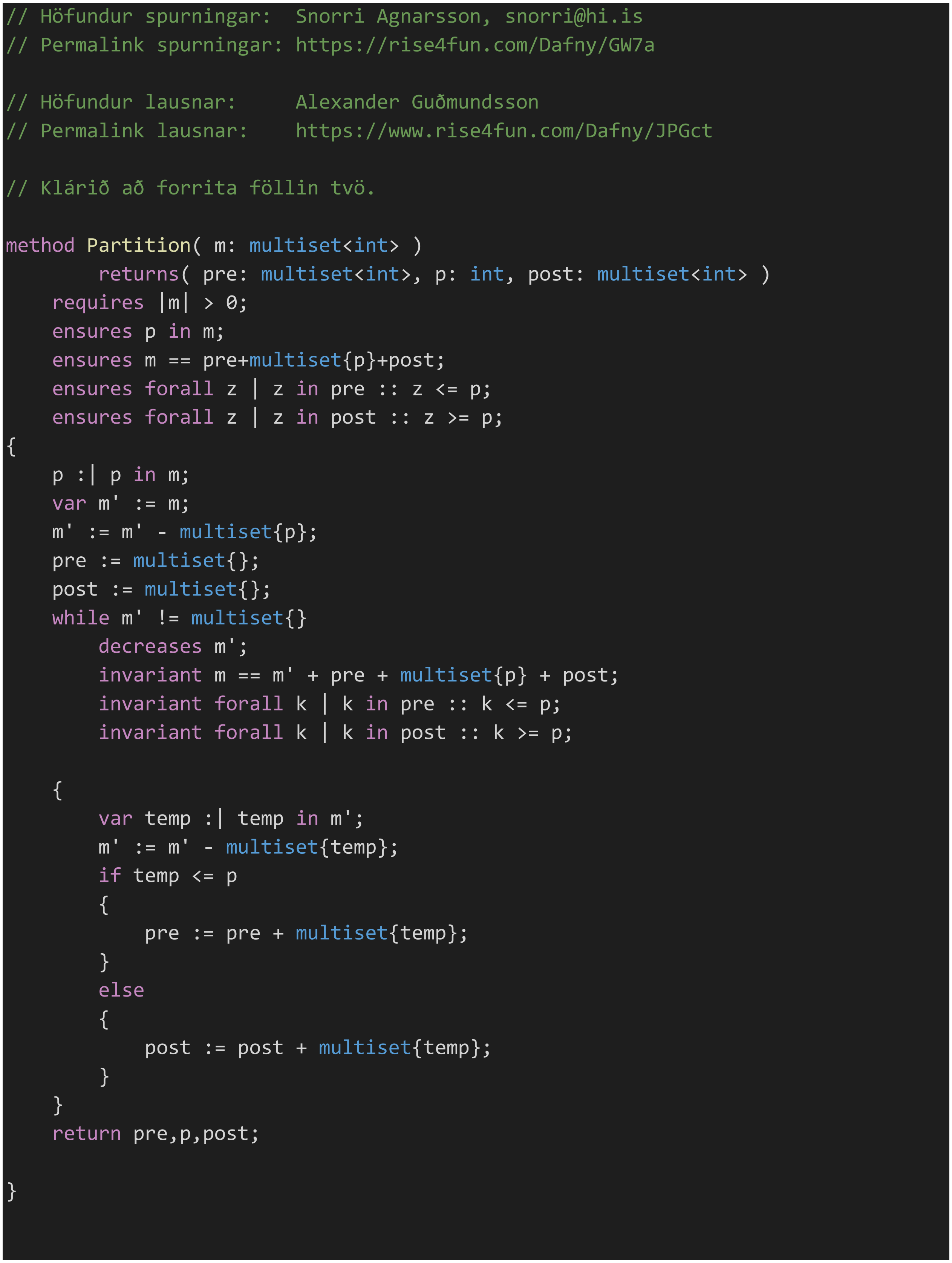
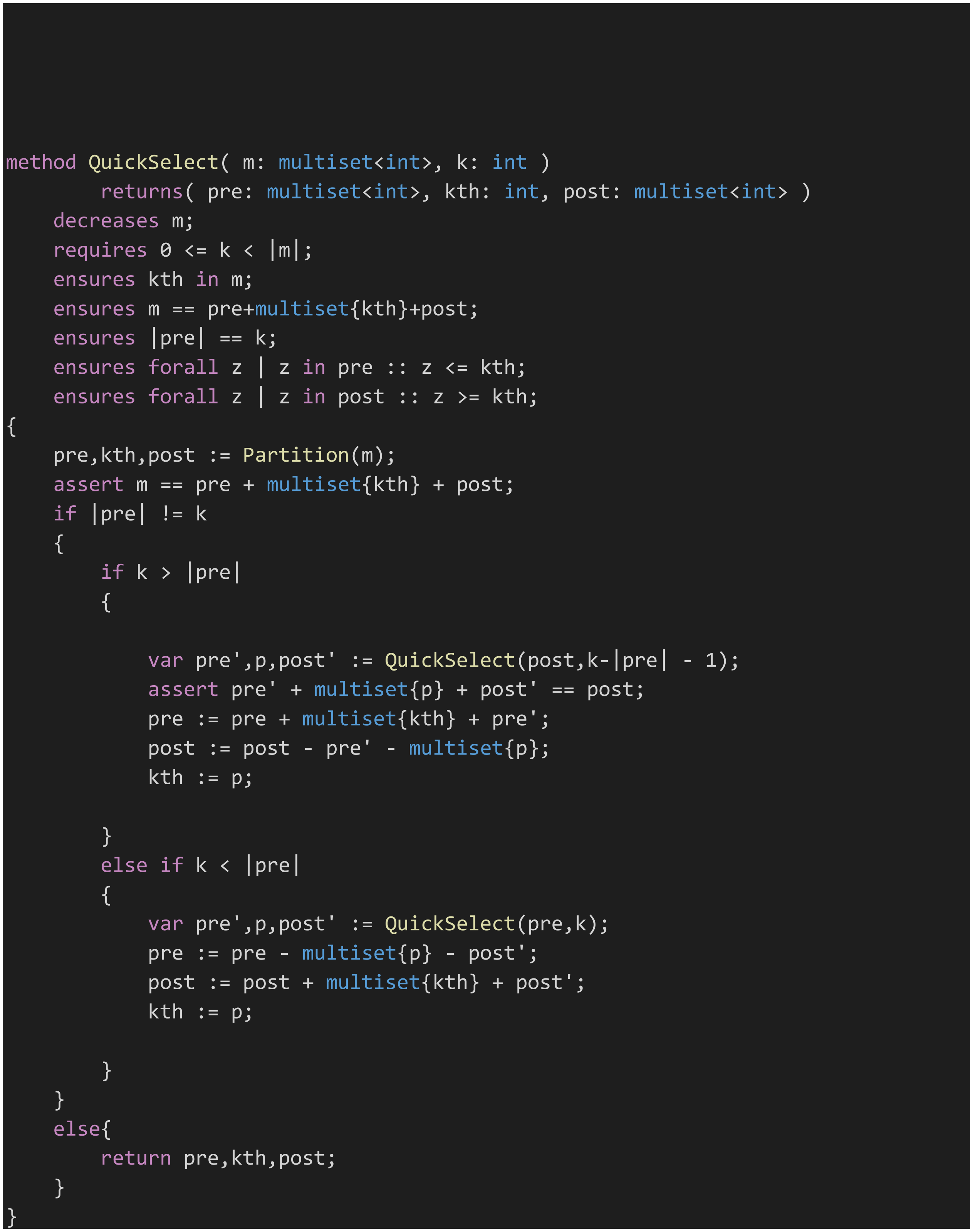
Alexander Guðmundsson
alg35@hi.is



```
// Höfundur spurningar:  Snorri Agnarsson, snorri@hi.is
// Permalink spurningar: https://rise4fun.com/Dafny/GW7a

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

// Klárið að forrita föllin tvö.

method Partition( m: multiset<int> )
        returns( pre: multiset<int>, p: int, post: multiset<int> )
    requires |m| > 0;
    ensures p in m;
    ensures m == pre+multiset{p}+post;
    ensures forall z | z in pre :: z <= p;
    ensures forall z | z in post :: z >= p;
{
    p :| p in m;
    var m' := m;
    m' := m' - multiset{p};
    pre := multiset{};
    post := multiset{};
    while m' != multiset{}
        decreases m';
        invariant m == m' + pre + multiset{p} + post;
        invariant forall k | k in pre :: k <= p;
        invariant forall k | k in post :: k >= p;

    {
        var temp :| temp in m';
        m' := m' - multiset{temp};
        if temp <= p
        {
            pre := pre + multiset{temp};
        }
        else
        {
            post := post + multiset{temp};
        }
    }
    return pre,p,post;

}
```

Alexander Guðmundsson
alg35@hi.is



```
method QuickSelect( m: multiset<int>, k: int )
        returns( pre: multiset<int>, kth: int, post: multiset<int> )
    decreases m;
    requires 0 <= k < |m|;
    ensures kth in m;
    ensures m == pre+multiset{kth}+post;
    ensures |pre| == k;
    ensures forall z | z in pre :: z <= kth;
    ensures forall z | z in post :: z >= kth;
{
    pre,kth,post := Partition(m);
    assert m == pre + multiset{kth} + post;
    if |pre| != k
    {
        if k > |pre|
        {

            var pre',p,post' := QuickSelect(post,k-|pre| - 1);
            assert pre' + multiset{p} + post' == post;
            pre := pre + multiset{kth} + pre';
            post := post - pre' - multiset{p};
            kth := p;

        }
        else if k < |pre|
        {
            var pre',p,post' := QuickSelect(pre,k);
            pre := pre - multiset{p} - post';
            post := post + multiset{kth} + post';
            kth := p;

        }
    }
    else{
        return pre,kth,post;
    }
}
```