# TÖL212M Rökstudd Forritun - Hópverkefni 1

Andri Fannar Kristjánsson

20. janúar 2025

## Hópverkefni 1

## 1

Klárið forritunina í skránni `LinearSearch-skeleton.dfy`, sem þið finnið í Canvas, á tvo vegu, með lykkju og með endurkvæmni. Ekki má breyta lýsingu fallsins sem skilgreint er að öðru leyti en því að leyfilegt er að bæta við „decreases“ klausu.

### 1.1 Svar:

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

```
method SearchRecursive( a: seq<int>, i: int, j: int, x: int )
    returns (k: int)
  requires 0 <= i <= j <= |a|
  ensures i <= k < j || k == -1
  ensures k != -1 ==> a[k] == x
  ensures k != -1 ==> forall r | k < r < j :: a[r] != x
  ensures k == -1 ==> forall r | i <= r < j :: a[r] != x
{
  // If all elements in the array have been checked, return -1
  // Works with lists of length 1 as i == j => 0 == 1 so -1 won't be
  // returned immediately
  if i == j
  {
    k := -1;
  }
  // Checks whether the element at index j-1 is equal to x
  // (since i <= k < j, we can't check j directly in the base case)
  else if a[j-1] == x
  {
    k := j-1;
  }
  // If the element at index j-1 is not equal to x, call the function
  // recursively with the last element removed
  else
  {
    k := SearchRecursive(a, i, j-1, x);
  }

  return;
}


method SearchLoop( a: seq<int>, i: int, j: int, x: int ) returns (k: int)
  requires 0 <= i <= j <= |a|
  ensures i <= k < j || k == -1
```

```
  ensures k != -1 ==> a[k] == x
  ensures k != -1 ==> forall r | k < r < j :: a[r] != x
  ensures k == -1 ==> forall r | i <= r < j :: a[r] != x
{
  // We start at the end of the array and move backwards, per method
  // description
  k := j - 1;

  while k >= i
    decreases k - i
    // Invariant states that all elements to the right of k are not
    // equal to x, as they have been checked
    // Fulfills the method's ensures clause
    invariant forall r :: k+1 <= r < j ==> a[r] != x
  {
    // Loop through and check each element in the array,
    // returning if it is equal to x
    if a[k] == x
    {
      return;
    }

    k := k - 1;
  }

  // If no element is equal to x, return -1
  k := -1;
  return;
}
```