

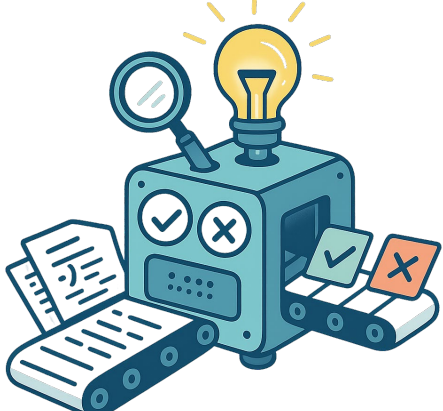
Query

Binary search:
Given a sorted array and a target
return the index if found else -1.

Code+Spec

```
method BinarySearch(a: array<int>, key: int) returns (idx: int)
  requires a != null
  requires forall i, j :: 0 <= i < j < a.Length ==> a[i] <= a[j]
  ensures idx == -1 ==> forall i :: 0 <= i < a.Length ==> a[i] != key
  ensures 0 <= idx < a.Length ==> a[idx] == key
{
  implementations here ...
}
```

Dafny Verifier



Hard-Coded

```
method Check_Spec_Strength(a: array<int>,key: int)returns (idx: int)
{
  idx := *;
  assume a!= null;
  assume forall i,j::0<=i<j< a.Length ==> a[i]<= a[j];
  assume idx==-1==>forall i::0<=i<a.length ==> a[i] != key;
  assume 0<=idx<a.Length==> a[idx]==key;
  var value :=BinarySearch(a, key);
  assert idx == value;
}
```

Dafny Verifier



Dafny
verified
Bidirectional
Proofs

⇒

Equiv.
Code &
Spec

Statement that Spec \implies Code

Generated Spec

```
method BinarySearch(a:array<int>,key:int)returns(idx: int)
  requires a!=null
  requires forall i,j::0<=i<j< a.Length ==> a[i]<= a[j]
  ensures idx==-1==>forall i::0<=i< a.Length ==> a[i] != key
  ensures 0<=idx<a.Length==>a[idx]== key
```

Natural Language Query

The method searches for key in a sorted, non-null arraya. If the key is not found, it returns negative one andguarantees the key is not in the array. If it finds the keyit returns the index of an element equal to the key

Validate the solution ✓