Experiment ‘p4_2_d_adjacent_duplicates’ Results

December 18, 2025

Experiment outcome: FAILURE_VERIFY
Bad responses: 0
Responses containing `assume` : 0
Resolution attempts: 8
Hard fails (resolution): 0
Soft fails (resolution): 2
Verification attempts: 6

Problem Specification

Problem name: p4_2_d_adjacent_duplicates
Natural language statement: Write a method that takes a sequence of integers and returns all adjacent duplicates. For example, if the input is [1, 3, 3, 4, 5, 5, 6, 6, 6, 2], the method should return [3, 5, 6].
Method signature: p4_2_d_adjacent_duplicates(inputs: seq<int>) returns (duplicates: seq<int>)

Ensures

- `forall d :: d in duplicates ==> d in inputs`
- `forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[i] in duplicates`
- `forall d :: d in duplicates ==> exists i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] && inputs[i] == d`
- `forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| && contiguous_duplicates(inpu` `i, j, d) == true`

Functional Code Given

```
function contiguous_duplicates(inputs: seq<int>, i : int, j : int, d : int) : (b : bool)
  requires 0 <= i < j < |inputs|
  decreases j - i
{
  if i + 1 == j then inputs[i] == inputs[j] == d
  else if inputs[i] == d then contiguous_duplicates(inputs, i + 1, j, d)
  else false
}
```

GenAI interactions

Below you will find all interactions between the ‘user’ (program) and the ‘assistant’ (OpenAI).

Program → GenAI

```
You are given the following task to perform in Dafny:

Write a method that takes a sequence of integers and returns all adjacent duplicates. For
    ↪  example, if the input is [1, 3, 3, 4, 5, 5, 6, 6, 6, 2], the method should return
    ↪  [3, 5, 6].

The signature should be:

method p4_2_d_adjacent_duplicates(inputs: seq<int>) returns (duplicates: seq<int>)

The method should respect the following contract:

ensures forall d :: d in duplicates ==> d in inputs, ensures forall i :: 0 <= i < |inputs
    ↪ | - 1 && inputs[i] == inputs[i + 1] ==> inputs[i] in duplicates, ensures forall d
    ↪ :: d in duplicates ==> exists i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i
    ↪ + 1] && inputs[i] == d, ensures forall d :: d in duplicates ==> exists i, j :: 0
    ↪ <= i < j < |inputs| && contiguous_duplicates(inputs, i, j, d) == true

The contract uses the following dafny code:

function contiguous_duplicates(inputs: seq<int>, i : int, j : int, d : int) : (b : bool)
  requires 0 <= i < j < |inputs|
  decreases j - i
{
  if i + 1 == j then inputs[i] == inputs[j] == d
  else if inputs[i] == d then contiguous_duplicates(inputs, i + 1, j, d)
  else false
}

These function(s) must not be used in your implementation of the method.

Produce and show only the Dafny body of this method, including the curly braces that
    ↪ surround it. Do not show the signature nor contract. You must not use 'assume'
    ↪ anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4bZZ0YxKQVdXFYLBtvzrkvMwRRk
Model: gpt-5.1-chat-latest
Created at: 1766050137
Finish reason: stop
Usage: 820 tokens in, and 251 tokens out

```
{
  duplicates := [];
  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < i && inputs[k] ==
        ↪ inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny verify, the below error is emitted and verify fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(12,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
12 |  ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[
    ↪ i] in duplicates
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^
```

```
problem_data/p4_2_d_adjacent_duplicates/program.dfy(13,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
13 |  ensures forall d :: d in duplicates ==> exists i :: 0 <= i < |inputs| - 1 && inputs
    ↪ [i] == inputs[i + 1] && inputs[i] == d
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(14,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
14 |  ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| &&
    ↪ contiguous_duplicates(inputs, i, j, d) == true
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(21,91): Error: index out of range
   |
21 | invariant forall d :: d in duplicates ==> exists k :: 0 <= k < i && inputs[k] ==
    ↪ inputs[k+1] && inputs[k] == d
   | ^


Dafny program verifier finished with 2 verified, 4 errors

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4beR9IW1f6d1mciMNXV3wMXjMYQ
Model: gpt-5.1-chat-latest
Created at: 1766050142
Finish reason: stop
Usage: 1710 tokens in, and 324 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs| - 1
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[
        ↪ k] == inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny verify, the below error is emitted and verify fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(12,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
12 |  ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[
    ↪ i] in duplicates
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(14,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
14 |  ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| &&
    ↪ contiguous_duplicates(inputs, i, j, d) == true
   | ^^^^^^
```

```
problem_data/p4_2_d_adjacent_duplicates/program.dfy(20,21): Error: this loop invariant
    ↪ could not be proved on entry
 Related message: loop invariant violation
   |
20 | invariant 0 <= i <= |inputs| - 1
   | ^^


Dafny program verifier finished with 2 verified, 3 errors

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4bjmvuXH4MzFImipdRv8EsKPCFN
Model: gpt-5.1-chat-latest
Created at: 1766050147
Finish reason: stop
Usage: 2216 tokens in, and 321 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[
        ↪ k] == inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny verify, the below error is emitted and verify fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(12,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
12 |  ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[
    ↪ i] in duplicates
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(14,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
14 |  ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| &&
    ↪ contiguous_duplicates(inputs, i, j, d) == true
   | ^^^^^^


Dafny program verifier finished with 2 verified, 2 errors

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4boORSdKMgY0zHk9TvVmu7darSG
Model: gpt-5.1-chat-latest
Created at: 1766050152
Finish reason: stop
Usage: 2736 tokens in, and 486 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[
        ↪ k] == inputs[k+1] && inputs[k] == d
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 &&
        ↪ contiguous_duplicates(inputs, k, k+1, d)
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny resolve, the below error is emitted and resolve fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(23,46): Warning: Could not find a
    ↪ trigger for this quantifier. Without a trigger, the quantifier may cause brittle
    ↪ verification. To silence this warning, add an explicit trigger using the {:trigger
    ↪ } attribute. For more information, see the section on quantifier instantiation
    ↪ rules in the reference manual.
   |
23 | invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 &&
    ↪ contiguous_duplicates(inputs, k, k+1, d)
   | ^^^^^^

Compilation failed because warnings were found and --allow-warnings is false

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4bvPpche52ZdIlgyl1Yo7e37PGN
Model: gpt-5.1-chat-latest
Created at: 1766050159
Finish reason: stop
Usage: 3108 tokens in, and 257 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[
        ↪ k] == inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny verify, the below error is emitted and verify fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(12,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
12 |  ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[i] in duplicates
   | ^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(14,9): Related location: this is the postcondition that could not be proved
   |
14 |  ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| && contiguous_duplicates(inputs, i, j, d) == true
   | ^^^^^^


Dafny program verifier finished with 2 verified, 2 errors

Correct the error by altering only the method body. Produce and show only the Dafny body, including the curly braces that surround it. Do not show the signature nor contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4bzjtKkrx0cyASawhh62biEtWgD
Model: gpt-5.1-chat-latest
Created at: 1766050163
Finish reason: stop
Usage: 3809 tokens in, and 294 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[k] == inputs[k+1] && inputs[k] == d
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && contiguous_duplicates(inputs, k, k+1, d)
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
    invariant 0 <= j <= |duplicates|
    invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
   {
    if duplicates[j] == d {
     found := true;
    }
    j := j + 1;
   }
   if !found {
    duplicates := duplicates + [d];
   }
  }
  i := i + 1;
 }
}
```

Program → GenAI

```
When using dafny resolve, the below error is emitted and resolve fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(23,46): Warning: Could not find a
    ↪ trigger for this quantifier. Without a trigger, the quantifier may cause brittle
    ↪ verification. To silence this warning, add an explicit trigger using the {:trigger
    ↪ } attribute. For more information, see the section on quantifier instantiation
    ↪ rules in the reference manual.
   |
23 | invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 &&
    ↪ contiguous_duplicates(inputs, k, k+1, d)
   | ^^^^^^

Compilation failed because warnings were found and --allow-warnings is false

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4c4iwgLv259JkZIgwzsM2x0JWVO
Model: gpt-5.1-chat-latest
Created at: 1766050168
Finish reason: stop
Usage: 4181 tokens in, and 385 tokens out

```
{
 duplicates := [];

 var i := 0;
 while i < |inputs| - 1
  invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < |inputs| - 1 && inputs[
        ↪ k] == inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Program → GenAI

```
When using dafny verify, the below error is emitted and verify fails:

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(12,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
12 |  ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs[
    ↪ i] in duplicates
   | ^^^^^^^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(15,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
15 | {
   | ^

problem_data/p4_2_d_adjacent_duplicates/program.dfy(14,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
14 |  ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| &&
    ↪ contiguous_duplicates(inputs, i, j, d) == true
   | ^^^^^^


Dafny program verifier finished with 2 verified, 2 errors

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪  including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:
ID: chatcmpl-Co4cAcAV4HWtAdD2bHQEB2GiQCJtk
Model: gpt-5.1-chat-latest
Created at: 1766050174
Finish reason: stop
Usage: 4882 tokens in, and 380 tokens out

```
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < i && inputs[k] ==
        ↪ inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Final Program

```
function contiguous_duplicates(inputs: seq<int>, i : int, j : int, d : int) : (b : bool)
  requires 0 <= i < j < |inputs|
  decreases j - i
{
  if i + 1 == j then inputs[i] == inputs[j] == d
  else if inputs[i] == d then contiguous_duplicates(inputs, i + 1, j, d)
  else false
}

method p4_2_d_adjacent_duplicates(inputs: seq<int>) returns (duplicates: seq<int>)
        ensures forall d :: d in duplicates ==> d in inputs
        ensures forall i :: 0 <= i < |inputs| - 1 && inputs[i] == inputs[i + 1] ==> inputs
            ↪ [i] in duplicates
        ensures forall d :: d in duplicates ==> exists i :: 0 <= i < |inputs| - 1 &&
            ↪ inputs[i] == inputs[i + 1] && inputs[i] == d
        ensures forall d :: d in duplicates ==> exists i, j :: 0 <= i < j < |inputs| &&
            ↪ contiguous_duplicates(inputs, i, j, d) == true
{
  duplicates := [];

  var i := 0;
  while i < |inputs| - 1
    invariant 0 <= i <= |inputs|
    invariant forall d :: d in duplicates ==> d in inputs
    invariant forall d :: d in duplicates ==> exists k :: 0 <= k < i && inputs[k] ==
        ↪ inputs[k+1] && inputs[k] == d
  {
    if inputs[i] == inputs[i+1] {
      var d := inputs[i];
      var found := false;
      var j := 0;
      while j < |duplicates|
        invariant 0 <= j <= |duplicates|
        invariant !found ==> forall k :: 0 <= k < j ==> duplicates[k] != d
      {
        if duplicates[j] == d {
          found := true;
        }
        j := j + 1;
      }
      if !found {
        duplicates := duplicates + [d];
      }
    }
    i := i + 1;
  }
}
```

Total Token Usage

Input tokens: 23462
Output tokens: 2698
Reasoning tokens: 576
Sum of 'total tokens': 26160

Experiment Timings

Iteration #8 started at 1766050176500, ended at 1766050183073, lasting 6573ms (6.57 seconds)
Overall Experiment started at 1766050139126, ended at 1766050183073, lasting 43947ms (43.95 seconds)
Iteration #4 started at 1766050154477, ended at 1766050160827, lasting 6350ms (6.35 seconds)
Iteration #5 started at 1766050160827, ended at 1766050165613, lasting 4786ms (4.79 seconds)
Iteration #6 started at 1766050165613, ended at 1766050170265, lasting 4652ms (4.65 seconds)
Iteration #7 started at 1766050170265, ended at 1766050176500, lasting 6235ms (6.24 seconds)
Iteration #1 started at 1766050139126, ended at 1766050144049, lasting 4923ms (4.92 seconds)
Iteration #2 started at 1766050144049, ended at 1766050149442, lasting 5393ms (5.39 seconds)
Iteration #3 started at 1766050149442, ended at 1766050154477, lasting 5035ms (5.04 seconds)