Experiment ‘p6_4_sum_without_smallest’ 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): 0
Verification attempts: 8

Problem Specification

Problem name: p6_4_sum_without_smallest
Natural language statement: Write a method that computes the sum of an array of values, except for the smallest one, in a single loop. In the loop, update the sum and the smallest value. After the loop, return the difference.
Method signature: p6_4_sum_without_smallest(arr: seq<int>) returns (sum: int)

Ensures

- `sum == sum_array(arr) - find_smallest(arr)`

Requires

- `|arr| >= 1`

Functional Code Given

```
function find_smallest(arr: seq<int>) : (ret : int)
  requires |arr| >= 1
  ensures forall x :: x in arr ==> ret <= x
{
  if |arr| == 1 then arr[0]
  else
    assert 2 <= |arr|;
    var rest_smallest := find_smallest(arr[1..]);
    var ret : int := if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    assert arr == [arr[0]] + arr[1..];
    assert ret == if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    ret
}

function sum_array(arr: seq<int>) : int
```

```
{
  if |arr| == 0 then 0
  else arr[0] + sum_array(arr[1..])
}
```

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 computes the sum of an array of values, except for the smallest one,
    ↪ in a single loop. In the loop, update the sum and the smallest value. After the
    ↪ loop, return the difference.

The signature should be:

method p6_4_sum_without_smallest(arr: seq<int>) returns (sum: int)

The method should respect the following contract:

requires |arr| >= 1, ensures sum == sum_array(arr) - find_smallest(arr)

The contract uses the following dafny code:

function find_smallest(arr: seq<int>) : (ret : int)
  requires |arr| >= 1
  ensures forall x :: x in arr ==> ret <= x
{
  if |arr| == 1 then arr[0]
  else
    assert 2 <= |arr|;
    var rest_smallest := find_smallest(arr[1..]);
    var ret : int := if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    assert arr == [arr[0]] + arr[1..];
    assert ret == if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    ret
}

function sum_array(arr: seq<int>) : int
{
  if |arr| == 0 then 0
  else arr[0] + sum_array(arr[1..])
}

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-Co5AG25A431wCaXw6EbRutkwrzyww
Model: gpt-5.1-chat-latest
Created at: 1766052288
Finish reason: stop
Usage: 754 tokens in, and 148 tokens out

```
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest == find_smallest(arr[..i])
  {
    runningSum := runningSum + arr[i];
    if arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(30,25): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
30 | invariant runningSum == sum_array(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,23): Error: this loop invariant
    ↪ could not be proved on entry
 Related message: loop invariant violation
   |
31 | invariant smallest == find_smallest(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,23): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
31 | invariant smallest == find_smallest(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,39): Error: function precondition
    ↪ could not be proved
   |
31 | invariant smallest == find_smallest(arr[..i])
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(2,17): Related location: this
    ↪ proposition could not be proved
  |
2 | requires |arr| >= 1
  | ^^


Dafny program verifier finished with 3 verified, 5 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-Co5AJ0UAJH9dxatprrmkCipH8CEkj
Model: gpt-5.1-chat-latest
Created at: 1766052291
Finish reason: stop
Usage: 1430 tokens in, and 214 tokens out

```
{
  var runningSum := arr[0];
  var smallest := arr[0];
  var i := 1;
  while i < |arr|
    invariant 1 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest == find_smallest(arr[..i])
  {
    runningSum := runningSum + arr[i];
    if arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(30,25): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
30 | invariant runningSum == sum_array(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,23): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
31 | invariant smallest == find_smallest(arr[..i])
   | ^^


Dafny program verifier finished with 3 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-Co5ANJmFavti0oB6hOLzbHG66nN9O
Model: gpt-5.1-chat-latest
Created at: 1766052295

Finish reason: stop
Usage: 1726 tokens in, and 276 tokens out

```
{
  var runningSum := arr[0];
  var smallest := arr[0];
  var i := 1;
  while i < |arr|
    invariant 1 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest == find_smallest(arr[..i])
  {
    // Update runningSum
    runningSum := runningSum + arr[i];
    assert runningSum == sum_array(arr[..i]) + arr[i];
    assert runningSum == sum_array(arr[..(i+1)]);

    // Update smallest
    if arr[i] < smallest {
      smallest := arr[i];
    }
    assert smallest == find_smallest(arr[..(i+1)]);

    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(36,22): Error: assertion might not
    ↪ hold
   |
36 | assert runningSum == sum_array(arr[..(i+1)]);
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(42,20): Error: assertion might not
    ↪ hold
   |
42 | assert smallest == find_smallest(arr[..(i+1)]);
   | ^^


Dafny program verifier finished with 3 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-Co5ATAtJPBRqOHONOaN3wQROYhsAt
Model: gpt-5.1-chat-latest
Created at: 1766052301
Finish reason: stop
Usage: 2185 tokens in, and 164 tokens out

```
{
  var runningSum := arr[0];
  var smallest := arr[0];
  var i := 1;
  while i < |arr|
    invariant 1 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest in arr[..i]
    invariant forall j :: 0 <= j < i ==> smallest <= arr[j]
  {
    runningSum := runningSum + arr[i];
    if arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(30,25): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
30 | invariant runningSum == sum_array(arr[..i])
   | ^^


Dafny program verifier finished with 3 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-Co5AXsBlgsPPNOwgYZTtfIVMMrYkB
Model: gpt-5.1-chat-latest
Created at: 1766052305
Finish reason: stop
Usage: 2550 tokens in, and 541 tokens out

```
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant (i == 0 ==> smallest == arr[0])
    invariant (i > 0 ==> smallest == find_smallest(arr[..i]))
  {
    runningSum := runningSum + arr[i];
    assert runningSum == sum_array(arr[..i]) + arr[i];
    assert runningSum == sum_array(arr[..(i+1)]);

    if arr[i] < smallest {
      smallest := arr[i];
    }
    assert smallest == find_smallest(arr[..(i+1)]);

    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(36,22): Error: assertion might not
    ↪ hold
   |
36 | assert runningSum == sum_array(arr[..(i+1)]);
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(41,20): Error: assertion might not
    ↪ hold
   |
41 | assert smallest == find_smallest(arr[..(i+1)]);
   | ^^


Dafny program verifier finished with 3 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-Co5AeiCettqRzNAWIuDSUFv4OzuRt
Model: gpt-5.1-chat-latest
Created at: 1766052312
Finish reason: stop
Usage: 3077 tokens in, and 349 tokens out

```
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest == find_smallest(arr[..i]) || i == 0
  {
    runningSum := runningSum + arr[i];
    if i == 0 || arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(30,25): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
30 | invariant runningSum == sum_array(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,39): Error: function precondition
    ↪ could not be proved
   |
31 | invariant smallest == find_smallest(arr[..i]) || i == 0
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(2,17): Related location: this
    ↪ proposition could not be proved
  |
2 | requires |arr| >= 1
  | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(31,50): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
31 | invariant smallest == find_smallest(arr[..i]) || i == 0
   | ^^
```

```
Dafny program verifier finished with 3 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-Co5AkfCJINkI5R49zqIAlkm4lXVGc
Model: gpt-5.1-chat-latest
Created at: 1766052318
Finish reason: stop
Usage: 3763 tokens in, and 246 tokens out

```
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant smallest in arr[..i] || i == 0
    invariant forall j :: 0 <= j < i ==> smallest <= arr[j]
  {
    runningSum := runningSum + arr[i];
    if i == 0 || arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  assert smallest == find_smallest(arr);
  sum := runningSum - smallest;
}
```

Program → GenAI

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

problem_data/p6_4_sum_without_smallest/program.dfy(24,0): Error: a postcondition could
    ↪ not be proved on this return path
   |
24 | {
   | ^

problem_data/p6_4_sum_without_smallest/program.dfy(23,13): Related location: this is the
    ↪ postcondition that could not be proved
   |
23 |  ensures sum == sum_array(arr) - find_smallest(arr)
```

```
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(30,25): Error: this invariant could
    ↪ not be proved to be maintained by the loop
 Related message: loop invariant violation
   |
30 | invariant runningSum == sum_array(arr[..i])
   | ^^

problem_data/p6_4_sum_without_smallest/program.dfy(40,18): Error: assertion might not
    ↪ hold
   |
40 | assert smallest == find_smallest(arr);
   | ^^


Dafny program verifier finished with 3 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-Co5AqaRaJhLIFqv8QZfipfYwfQOhp
Model: gpt-5.1-chat-latest
Created at: 1766052324
Finish reason: stop
Usage: 4101 tokens in, and 414 tokens out

```
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant i == 0 || smallest == find_smallest(arr[..i])
  {
    runningSum := runningSum + arr[i];
    if i == 0 || arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Final Program

```
function find_smallest(arr: seq<int>) : (ret : int)
  requires |arr| >= 1
  ensures forall x :: x in arr ==> ret <= x
{
  if |arr| == 1 then arr[0]
  else
    assert 2 <= |arr|;
    var rest_smallest := find_smallest(arr[1..]);
    var ret : int := if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    assert arr == [arr[0]] + arr[1..];
    assert ret == if arr[0] <= rest_smallest then arr[0] else rest_smallest;
    ret
}

function sum_array(arr: seq<int>) : int
{
  if |arr| == 0 then 0
  else arr[0] + sum_array(arr[1..])
}

method p6_4_sum_without_smallest(arr: seq<int>) returns (sum: int)
       requires |arr| >= 1
       ensures sum == sum_array(arr) - find_smallest(arr)
{
  var runningSum := 0;
  var smallest := arr[0];
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant runningSum == sum_array(arr[..i])
    invariant i == 0 || smallest == find_smallest(arr[..i])
  {
    runningSum := runningSum + arr[i];
    if i == 0 || arr[i] < smallest {
      smallest := arr[i];
    }
    i := i + 1;
  }
  sum := runningSum - smallest;
}
```

Total Token Usage

Input tokens: 19586
Output tokens: 2352
Reasoning tokens: 960
Sum of ‘total tokens’: 21938

Experiment Timings

Iteration #8 started at 1766052325967, ended at 1766052333537, lasting 7570ms (7.57 seconds)
Overall Experiment started at 1766052290021, ended at 1766052333537, lasting 43516ms (43.52 seconds)
Iteration #4 started at 1766052302806, ended at 1766052307195, lasting 4389ms (4.39 seconds)
Iteration #5 started at 1766052307195, ended at 1766052314693, lasting 7498ms (7.50 seconds)
Iteration #6 started at 1766052314693, ended at 1766052320755, lasting 6062ms (6.06 seconds)
Iteration #7 started at 1766052320755, ended at 1766052325967, lasting 5212ms (5.21 seconds)
Iteration #1 started at 1766052290021, ended at 1766052293496, lasting 3475ms (3.48 seconds)
Iteration #2 started at 1766052293496, ended at 1766052297675, lasting 4179ms (4.18 seconds)
Iteration #3 started at 1766052297675, ended at 1766052302806, lasting 5131ms (5.13 seconds)