

Experiment ‘p6_2_h_is_sorted’ 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_2_h_is_sorted

Natural language statement: Write a method to return true if the given array is currently sorted in increasing order.

Method signature: p6_2_h_is_sorted(arr: seq<int>) returns (isSorted: bool)

Ensures

- `isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 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 to return true if the given array is currently sorted in increasing order.

The signature should be:

```
method p6_2_h_is_sorted(arr: seq<int>) returns (isSorted: bool)
```

The method should respect the following contract:

```
ensures isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 1])
```

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-Co58I3HwrFdV2zCbMdBOHJsGVWRAO

Model: gpt-5.1-chat-latest

Created at: 1766052166

Finish reason: stop

Usage: 304 tokens in, and 328 tokens out

```
{
    var sorted := true;
    var i := 0;
    while i < |arr| - 1
        invariant 0 <= i <= |arr| - 1
        invariant sorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
    {
        if arr[i] > arr[i + 1] {
            sorted := false;
            break;
        }
        i := i + 1;
    }
    isSorted := sorted;
}
```

Program → GenAI

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

```

problem_data/p6_2_h_is_sorted/program.dfy(7,21): Error: this loop invariant could not be
    ↵ proved on entry
Related message: loop invariant violation
|
7 | invariant 0 <= i <= |arr| - 1
| ^

Dafny program verifier finished with 1 verified, 1 error

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: chatmpl-Co58Nj0YOIzEnQwZR46GzK1D45FOB
Model: gpt-5.1-chat-latest
Created at: 1766052171
Finish reason: stop
Usage: 581 tokens in, and 197 tokens out

```
{
    var sorted := true;
    var i := 0;
    while i < |arr| - 1
        invariant 0 <= i <= |arr|
        invariant sorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
    {
        if arr[i] > arr[i + 1] {
            sorted := false;
            break;
        }
        i := i + 1;
    }
    isSorted := sorted;
}
```

Program → GenAI

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

```

problem_data/p6_2_h_is_sorted/program.dfy(8,66): Error: index out of range
|
8 | invariant sorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
| ^

```

Dafny program verifier finished with 1 verified, 1 error

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-Co58SXPTSDGhmFWdFP3TYIqBZGJtK

Model: gpt-5.1-chat-latest

Created at: 1766052176

Finish reason: stop

Usage: 850 tokens in, and 200 tokens out

```
{
    var sorted := true;
    var i := 0;
    while i < |arr| - 1
        invariant 0 <= i <= |arr| - 1
        invariant sorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
    {
        if arr[i] > arr[i + 1] {
            sorted := false;
            break;
        }
        i := i + 1;
    }
    isSorted := sorted;
}
```

Program → GenAI

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

```
problem_data/p6_2_h_is_sorted/program.dfy(7,21): Error: this loop invariant could not be
  ↪ proved on entry
Related message: loop invariant violation
|
7 | invariant 0 <= i <= |arr| - 1
| ^~
```

Dafny program verifier finished with 1 verified, 1 error

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-Co58VuV8zTU3FGoEItG6yCExJPdUr

Model: gpt-5.1-chat-latest
Created at: 1766052179
Finish reason: stop
Usage: 1125 tokens in, and 392 tokens out

```
{  
    var sorted := true;  
    var i := 0;  
    while i < |arr| - 1  
        invariant 0 <= i <= |arr|  
        invariant sorted ==> (forall j :: 0 <= j < i - 1 ==> arr[j] <= arr[j + 1])  
    {  
        if arr[i] > arr[i + 1] {  
            sorted := false;  
            break;  
        }  
        i := i + 1;  
    }  
    isSorted := sorted;  
}
```

Program → GenAI

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

```
problem_data/p6_2_h_is_sorted/program.dfy(3,0): Error: a postcondition could not be  
    ↪ proved on this return path  
|  
3 | {  
| ^  
  
problem_data/p6_2_h_is_sorted/program.dfy(2,18): Related location: this is the  
    ↪ postcondition that could not be proved  
|  
2 | ensures isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 1])  
| ^^  
  
problem_data/p6_2_h_is_sorted/program.dfy(8,26): Error: this invariant could not be  
    ↪ proved to be maintained by the loop  
Related message: loop invariant violation  
|  
8 | invariant sorted ==> (forall j :: 0 <= j < i - 1 ==> arr[j] <= arr[j + 1])  
| ~~~~~
```

Dafny program verifier finished with 1 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-Co58blOdPnWyfLTHVSQG9ldhJWyo

Model: gpt-5.1-chat-latest

Created at: 1766052185

Finish reason: stop

Usage: 1661 tokens in, and 137 tokens out

```
{  
    isSorted := true;  
    for i := 0 to |arr| - 2 {  
        if arr[i] > arr[i + 1] {  
            isSorted := false;  
            return;  
        }  
    }  
}
```

Program → GenAI

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

```
problem_data/p6_2_h_is_sorted/program.dfy(3,0): Error: a postcondition could not be  
    ↪ proved on this return path  
|  
3 | {  
| ^  
  
problem_data/p6_2_h_is_sorted/program.dfy(2,18): Related location: this is the  
    ↪ postcondition that could not be proved  
|  
2 |   ensures isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 1])  
| ^  
  
problem_data/p6_2_h_is_sorted/program.dfy(5,11): Error: lower bound must not exceed upper  
    ↪ bound  
|  
5 | for i := 0 to |arr| - 2 {  
| ^
```

Dafny program verifier finished with 1 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-Co58fdUQOZCCeRRL52OtDUv8nELXH

Model: gpt-5.1-chat-latest
Created at: 1766052189
Finish reason: stop
Usage: 1947 tokens in, and 320 tokens out

```
{  
    isSorted := true;  
    var i := 0;  
    while i < |arr| - 1  
        invariant 0 <= i <= |arr|  
        invariant isSorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])  
    {  
        if arr[i] > arr[i + 1] {  
            isSorted := false;  
            break;  
        }  
        i := i + 1;  
    }  
}
```

Program → GenAI

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

```
problem_data/p6_2_h_is_sorted/program.dfy(8,68): Error: index out of range  
|  
8 | invariant isSorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])  
| ^
```

Dafny program verifier finished with 1 verified, 1 error

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-Co58lP3RW8O6b38Qg1AC2aylTSrC4
Model: gpt-5.1-chat-latest
Created at: 1766052195
Finish reason: stop
Usage: 2110 tokens in, and 131 tokens out

```
{  
    isSorted := true;  
    var i := 0;  
    while i < |arr| - 1  
        invariant 0 <= i <= |arr|  
        invariant isSorted ==> (forall j :: 0 <= j < i - 1 ==> arr[j] <= arr[j + 1])
```

```

{
  if arr[i] > arr[i + 1] {
    isSorted := false;
    break;
  }
  i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p6_2_h_is_sorted/program.dfy(3,0): Error: a postcondition could not be
  ↪ proved on this return path
|
3 | {
| ^
|
problem_data/p6_2_h_is_sorted/program.dfy(2,18): Related location: this is the
  ↪ postcondition that could not be proved
|
2 | ensures isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 1])
| ^^

problem_data/p6_2_h_is_sorted/program.dfy(8,28): Error: this invariant could not be
  ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
8 | invariant isSorted ==> (forall j :: 0 <= j < i - 1 ==> arr[j] <= arr[j + 1])
| ~~~~~

```

Dafny program verifier finished with 1 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: chatmpl-Co58oVl5KYOjEwJm7WlTgsQ0P3JLf

Model: gpt-5.1-chat-latest

Created at: 1766052198

Finish reason: stop

Usage: 2643 tokens in, and 131 tokens out

```

{
  isSorted := true;
  var i := 0;
}
}

```

```

while i < |arr| - 1
    invariant 0 <= i <= |arr| - 1
    invariant isSorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
{
    if arr[i] > arr[i + 1] {
        isSorted := false;
        break;
    }
    i := i + 1;
}
}

```

Final Program

```

method p6_2_h_is_sorted(arr: seq<int>) returns (isSorted: bool)
    ensures isSorted == (forall i :: 0 <= i < |arr| - 1 ==> arr[i] <= arr[i + 1])
{
    isSorted := true;
    var i := 0;
    while i < |arr| - 1
        invariant 0 <= i <= |arr| - 1
        invariant isSorted ==> (forall j :: 0 <= j < i ==> arr[j] <= arr[j + 1])
    {
        if arr[i] > arr[i + 1] {
            isSorted := false;
            break;
        }
        i := i + 1;
    }
}

```

Total Token Usage

Input tokens: 11221
Output tokens: 1836
Reasoning tokens: 832
Sum of ‘total tokens’: 13057

Experiment Timings

Iteration #8 started at 1766052200818, ended at 1766052204803, lasting 3985ms (3.99 seconds)
Overall Experiment started at 1766052168452, ended at 1766052204803, lasting 36351ms (36.35 seconds)
Iteration #4 started at 1766052181632, ended at 1766052187437, lasting 5805ms (5.81 seconds)
Iteration #5 started at 1766052187437, ended at 1766052191124, lasting 3687ms (3.69 seconds)
Iteration #6 started at 1766052191124, ended at 1766052197303, lasting 6179ms (6.18 seconds)
Iteration #7 started at 1766052197303, ended at 1766052200818, lasting 3515ms (3.52 seconds)
Iteration #1 started at 1766052168452, ended at 1766052173433, lasting 4981ms (4.98 seconds)
Iteration #2 started at 1766052173433, ended at 1766052177538, lasting 4105ms (4.11 seconds)
Iteration #3 started at 1766052177538, ended at 1766052181632, lasting 4094ms (4.09 seconds)