

Experiment ‘p4_5_analyze_floats’ 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): 1

Verification attempts: 7

Problem Specification

Problem name: p4_5_analyze_floats

Natural language statement: Write a method that takes a sequence of floatingpoint numbers and returns:
- the average of the values. - the smallest of the values. - the largest of the values. - the range, that is the difference between the smallest and largest.

Method signature: p4_5_analyze_floats(values: seq<real>) returns (average: real, min: real, max: real, range: real)

Ensures

- `average == (sum(values) / |values| as real)`
- `forall v :: v in values ==> min <= v`
- `forall v :: v in values ==> max >= v`
- `range == max - min`
- `exists v :: v in values && min == v`
- `exists v :: v in values && max == v`

Requires

- `0 < |values|`

Functional Code Given

```
function sum(values: seq<real>): real
{
  if |values| == 0 then 0.0
  else values[0] + sum(values[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 takes a sequence of floatingpoint numbers and returns: - the average
→ of the values. - the smallest of the values. - the largest of the values. - the
→ range, that is the difference between the smallest and largest.

The signature should be:

```
method p4_5_analyze_floats(values: seq<real>) returns (average: real, min: real, max:
→ real, range: real)
```

The method should respect the following contract:

```
requires 0 < |values|, ensures average == (sum(values) / |values| as real), ensures
→ forall v :: v in values ==> min <= v, ensures forall v :: v in values ==> max >= v
→ , ensures range == max - min, ensures exists v :: v in values && min == v, ensures
→ exists v :: v in values && max == v
```

The contract uses the following dafny code:

```
function sum(values: seq<real>): real
{
  if |values| == 0 then 0.0
  else values[0] + sum(values[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-Co4e6sDiKkkaL5GGvtVoYsP9xvswa

Model: gpt-5.1-chat-latest

Created at: 1766050294

Finish reason: stop

Usage: 614 tokens in, and 318 tokens out

```
{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
```

```

while i < |values|
  invariant 0 <= i <= |values|
  invariant s == sum(values[..i])
  invariant forall v :: v in values[..i] ==> min <= v
  invariant forall v :: v in values[..i] ==> max >= v
  invariant (exists v :: v in values[..i] && min == v) || i == 0
  invariant (exists v :: v in values[..i] && max == v) || i == 0
{
  s := s + values[i];
  if values[i] < min {
    min := values[i];
  }
  if values[i] > max {
    max := values[i];
  }
  i := i + 1;
}
average := s / (|values| as real);
range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↪ proved on this return path

```

```

|
15 | {
| ^

```

```

problem_data/p4_5_analyze_floats/program.dfy(9,17): Related location: this is the
  ↪ postcondition that could not be proved

```

```

|
9 | ensures average == (sum(values) / |values| as real)
| ^^

```

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↪ proved on this return path

```

```

|
15 | {
| ^

```

```

problem_data/p4_5_analyze_floats/program.dfy(10,9): Related location: this is the
  ↪ postcondition that could not be proved

```

```

|
10 | ensures forall v :: v in values ==> min <= v
| ~~~~~

```

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↪ proved on this return path

```

```

15 | {
    | ~

problem_data/p4_5_analyze_floats/program.dfy(11,9): Related location: this is the
    ↪ postcondition that could not be proved
    |
11 | ensures forall v :: v in values ==> max >= v
    | ~~~~~~

problem_data/p4_5_analyze_floats/program.dfy(25,57): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
    |
25 | invariant (exists v :: v in values[..i] && min == v) || i == 0
    | ^^

problem_data/p4_5_analyze_floats/program.dfy(26,57): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
    |
26 | invariant (exists v :: v in values[..i] && max == v) || i == 0
    | ^^

Dafny program verifier finished with 2 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-Co4eBj4Q1TNVbdljwZBLOmFxcSZdq

Model: gpt-5.1-chat-latest

Created at: 1766050299

Finish reason: stop

Usage: 1626 tokens in, and 250 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant exists v :: v in values[..i] ==> min == v || i == 0

```

```

    invariant exists v :: v in values[..i] ==> max == v || i == 0
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
    i := i + 1;
  }
  average := s / (|values| as real);
  range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(25,14): Warning: the quantifier has the form
  ↳ 'exists x :: A ==> B', which most often is a typo for 'exists x :: A && B'; if
  ↳ you think otherwise, rewrite as 'exists x :: (A ==> B)' or 'exists x :: !A || B'
  ↳ to suppress this warning

```

```

|
25 | invariant exists v :: v in values[..i] ==> min == v || i == 0
| ~~~~~

```

```

problem_data/p4_5_analyze_floats/program.dfy(26,14): Warning: the quantifier has the form
  ↳ 'exists x :: A ==> B', which most often is a typo for 'exists x :: A && B'; if
  ↳ you think otherwise, rewrite as 'exists x :: (A ==> B)' or 'exists x :: !A || B'
  ↳ to suppress this warning

```

```

|
26 | invariant exists v :: v in values[..i] ==> max == v || i == 0
| ~~~~~

```

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-Co4eFUj7kxC7TgCwADe1la8IwbUnn

Model: gpt-5.1-chat-latest

Created at: 1766050303

Finish reason: stop

Usage: 1956 tokens in, and 318 tokens out

```

{

```

```

var s: real := 0.0;
min := values[0];
max := values[0];
var i: int := 0;
while i < |values|
  invariant 0 <= i <= |values|
  invariant s == sum(values[..i])
  invariant forall v :: v in values[..i] ==> min <= v
  invariant forall v :: v in values[..i] ==> max >= v
  invariant (exists v :: v in values[..i] && min == v) || i == 0
  invariant (exists v :: v in values[..i] && max == v) || i == 0
{
  s := s + values[i];
  if values[i] < min {
    min := values[i];
  }
  if values[i] > max {
    max := values[i];
  }
  i := i + 1;
}
average := s / (|values| as real);
range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↳ proved on this return path
|
15 | {
| ^

problem_data/p4_5_analyze_floats/program.dfy(9,17): Related location: this is the
  ↳ postcondition that could not be proved
|
9 | ensures average == (sum(values) / |values| as real)
| ^^

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↳ proved on this return path
|
15 | {
| ^

problem_data/p4_5_analyze_floats/program.dfy(10,9): Related location: this is the
  ↳ postcondition that could not be proved
|
10 | ensures forall v :: v in values ==> min <= v

```

```

    | ~~~~~~

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
    ↪ proved on this return path
    |
15 | {
    | ~

problem_data/p4_5_analyze_floats/program.dfy(11,9): Related location: this is the
    ↪ postcondition that could not be proved
    |
11 | ensures forall v :: v in values ==> max >= v
    | ~~~~~~

problem_data/p4_5_analyze_floats/program.dfy(25,57): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
    |
25 | invariant (exists v :: v in values[..i] && min == v) || i == 0
    | ^^

problem_data/p4_5_analyze_floats/program.dfy(26,57): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
    |
26 | invariant (exists v :: v in values[..i] && max == v) || i == 0
    | ^^

Dafny program verifier finished with 2 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-Co4eLK1XqWMrQTDgpQhESkB22BqPg

Model: gpt-5.1-chat-latest

Created at: 1766050309

Finish reason: stop

Usage: 2956 tokens in, and 299 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|

```

```

invariant s == sum(values[..i])
invariant i == 0 ==> min == values[0]
invariant i == 0 ==> max == values[0]
invariant i > 0 ==> forall v :: v in values[..i] ==> min <= v
invariant i > 0 ==> forall v :: v in values[..i] ==> max >= v
invariant i > 0 ==> exists v :: v in values[..i] && min == v
invariant i > 0 ==> exists v :: v in values[..i] && max == v
{
  s := s + values[i];
  if i > 0 {
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
  }
  i := i + 1;
}
average := s / (|values| as real);
range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
  ↳ proved on this return path

```

```

|
15 | {
| ~

```

```

problem_data/p4_5_analyze_floats/program.dfy(9,17): Related location: this is the
  ↳ postcondition that could not be proved

```

```

|
9 | ensures average == (sum(values) / |values| as real)
| ^^

```

```

problem_data/p4_5_analyze_floats/program.dfy(22,16): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation

```

```

|
22 | invariant s == sum(values[..i])
| ^^

```

```

problem_data/p4_5_analyze_floats/program.dfy(25,24): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation

```

```

|
25 | invariant i > 0 ==> forall v :: v in values[..i] ==> min <= v

```



```

| ~~~~~~

problem_data/p4_5_analyze_floats/program.dfy(26,24): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation
|
26 | invariant i > 0 ==> forall v :: v in values[..i] ==> max >= v
| ~~~~~~

problem_data/p4_5_analyze_floats/program.dfy(27,24): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation
|
27 | invariant i > 0 ==> exists v :: v in values[..i] && min == v
| ~~~~~~

Dafny program verifier finished with 2 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-Co4eQpZtmr5AwJWPSOsksp4jDeuv9

Model: gpt-5.1-chat-latest

Created at: 1766050314

Finish reason: stop

Usage: 3639 tokens in, and 234 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant exists v :: v in values[..i] && min == v
    invariant exists v :: v in values[..i] && max == v
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
  }
}

```

```

    }
    i := i + 1;
  }
  average := s / (|values| as real);
  range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(25,14): Error: this loop invariant could not
  → be proved on entry
  Related message: loop invariant violation
  |
25 | invariant exists v :: v in values[..i] && min == v
  | ~~~~~

```

Dafny program verifier finished with 2 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-Co4eUHuGFUMZIHOB39JePgyDeX2nt

Model: gpt-5.1-chat-latest

Created at: 1766050318

Finish reason: stop

Usage: 3713 tokens in, and 250 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant i == 0 || exists v :: v in values[..i] && min == v
    invariant i == 0 || exists v :: v in values[..i] && max == v
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
  }
}

```

```

    if values[i] > max {
      max := values[i];
    }
    i := i + 1;
  }
  average := s / (|values| as real);
  range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
    ↪ proved on this return path

```

```

|
15 | {
| ~

```

```

problem_data/p4_5_analyze_floats/program.dfy(9,17): Related location: this is the
    ↪ postcondition that could not be proved

```

```

|
9 | ensures average == (sum(values) / |values| as real)
| ^^

```

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
    ↪ proved on this return path

```

```

|
15 | {
| ~

```

```

problem_data/p4_5_analyze_floats/program.dfy(10,9): Related location: this is the
    ↪ postcondition that could not be proved

```

```

|
10 | ensures forall v :: v in values ==> min <= v
| ~~~~~

```

```

problem_data/p4_5_analyze_floats/program.dfy(15,0): Error: a postcondition could not be
    ↪ proved on this return path

```

```

|
15 | {
| ~

```

```

problem_data/p4_5_analyze_floats/program.dfy(11,9): Related location: this is the
    ↪ postcondition that could not be proved

```

```

|
11 | ensures forall v :: v in values ==> max >= v
| ~~~~~

```

```

problem_data/p4_5_analyze_floats/program.dfy(25,21): Error: this invariant could not be
    ↪ proved to be maintained by the loop

```

```

Related message: loop invariant violation
|
25 | invariant i == 0 || exists v :: v in values[..i] && min == v
| ^^

problem_data/p4_5_analyze_floats/program.dfy(26,21): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation
|
26 | invariant i == 0 || exists v :: v in values[..i] && max == v
| ^^

Dafny program verifier finished with 2 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-Co4eZQ0C6cCQXnopCS0Vv367WcL7w

Model: gpt-5.1-chat-latest

Created at: 1766050323

Finish reason: stop

Usage: 4859 tokens in, and 234 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant exists v :: v in values[..i] && min == v
    invariant exists v :: v in values[..i] && max == v
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
    i := i + 1;
  }
  average := s / (|values| as real);
}

```

```

    range := max - min;
}

```

Program → GenAI

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

```

problem_data/p4_5_analyze_floats/program.dfy(25,14): Error: this loop invariant could not
    ↪ be proved on entry

```

Related message: loop invariant violation

```

|
25 | invariant exists v :: v in values[..i] && min == v
| ~~~~~

```

Dafny program verifier finished with 2 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-Co4ed3ApiXSQnaziWewuw5qNT03Is

Model: gpt-5.1-chat-latest

Created at: 1766050327

Finish reason: stop

Usage: 4867 tokens in, and 314 tokens out

```

{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant i == 0 || exists v :: v in values[..i] && min == v
    invariant i == 0 || exists v :: v in values[..i] && max == v
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
    i := i + 1;
  }
}

```

```

}
average := s / (|values| as real);
range := max - min;
}

```

Final Program

```

function sum(values: seq<real>): real
{
  if |values| == 0 then 0.0
  else values[0] + sum(values[1..])
}

method p4_5_analyze_floats(values: seq<real>) returns (average: real, min: real, max:
  ↪ real, range: real)
  requires 0 < |values|
  ensures average == (sum(values) / |values| as real)
  ensures forall v :: v in values ==> min <= v
  ensures forall v :: v in values ==> max >= v
  ensures range == max - min
  ensures exists v :: v in values && min == v
  ensures exists v :: v in values && max == v
{
  var s: real := 0.0;
  min := values[0];
  max := values[0];
  var i: int := 0;
  while i < |values|
    invariant 0 <= i <= |values|
    invariant s == sum(values[..i])
    invariant forall v :: v in values[..i] ==> min <= v
    invariant forall v :: v in values[..i] ==> max >= v
    invariant i == 0 || exists v :: v in values[..i] && min == v
    invariant i == 0 || exists v :: v in values[..i] && max == v
  {
    s := s + values[i];
    if values[i] < min {
      min := values[i];
    }
    if values[i] > max {
      max := values[i];
    }
    i := i + 1;
  }
  average := s / (|values| as real);
  range := max - min;
}

```

Total Token Usage

Input tokens: 24230

Output tokens: 2217

Reasoning tokens: 192

Sum of ‘total tokens’: 26447

Experiment Timings

Iteration #8 started at 1766050329519, ended at 1766050334537, lasting 5018ms (5.02 seconds)

Overall Experiment started at 1766050296192, ended at 1766050334537, lasting 38345ms (38.35 seconds)

Iteration #4 started at 1766050311058, ended at 1766050316370, lasting 5312ms (5.31 seconds)

Iteration #5 started at 1766050316370, ended at 1766050320738, lasting 4368ms (4.37 seconds)

Iteration #6 started at 1766050320738, ended at 1766050325531, lasting 4793ms (4.79 seconds)

Iteration #7 started at 1766050325531, ended at 1766050329519, lasting 3988ms (3.99 seconds)

Iteration #1 started at 1766050296193, ended at 1766050301322, lasting 5129ms (5.13 seconds)

Iteration #2 started at 1766050301322, ended at 1766050305596, lasting 4274ms (4.27 seconds)

Iteration #3 started at 1766050305597, ended at 1766050311058, lasting 5461ms (5.46 seconds)