

# Experiment ‘p6\_10\_same\_set’ Results

December 18, 2025

**Experiment outcome:** SUCCESS

**Bad responses:** 0

**Responses containing assume :** 0

**Resolution attempts:** 4

**Hard fails (resolution):** 2

**Soft fails (resolution):** 0

**Verification attempts:** 2

## Problem Specification

**Problem name:** p6\_10\_same\_set

**Natural language statement:** Write a method that checks whether two arrays have the same elements in some order, ignoring duplicates.

**Method signature:** p6\_10\_same\_set(arr1: seq<int>, arr2: seq<int>) returns (areSameSet: bool)

### Ensures

- areSameSet == (array\_to\_set(arr1) == array\_to\_set(arr2))

### Functional Code Given

```
function array_to_set(arr: seq<int>) : set<int>
{
  if |arr| == 0 then {}
  else {arr[0]} + array_to_set(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 checks whether two arrays have the same elements in some order,  
↳ ignoring duplicates.

The signature should be:

```
method p6_10_same_set(arr1: seq<int>, arr2: seq<int>) returns (areSameSet: bool)
```

The method should respect the following contract:

```
ensures areSameSet == (array_to_set(arr1) == array_to_set(arr2))
```

The contract uses the following dafny code:

```
function array_to_set(arr: seq<int>) : set<int>
{
  if |arr| == 0 then {}
  else {arr[0]} + array_to_set(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: chatmpl-Co5BoQdBVPh5OMIn0eHxUIvOAgQcC

Model: gpt-5.1-chat-latest

Created at: 1766052384

Finish reason: stop

Usage: 430 tokens in, and 269 tokens out

```
{
  var s1: set<int> := {};
  var i := 0;
  while i < |arr1|
    invariant 0 <= i <= |arr1|
    invariant s1 == set arr1[0..i]
  {
    s1 := s1 + {arr1[i]};
    i := i + 1;
  }
}
```

```

var s2: set<int> := {};
var j := 0;
while j < |arr2|
    invariant 0 <= j <= |arr2|
    invariant s2 == set arr2[0..j]
{
    s2 := s2 + {arr2[j]};
    j := j + 1;
}

areSameSet := (s1 == s2);
}

```

## Program → GenAI

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

```

problem_data/p6_10_same_set/program.dfy(14,28): Error: invalid WhileStmt
|
14 | invariant s1 == set arr1[0..i]
| ^
1 parse errors detected in program.dfy

```

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-Co5BsOJVraMx7Hgp0l5ZMjIMwEnR2

**Model:** gpt-5.1-chat-latest

**Created at:** 1766052388

**Finish reason:** stop

**Usage:** 671 tokens in, and 335 tokens out

```
{
var s1: set<int> := {};
var i := 0;
while i < |arr1|
    invariant 0 <= i <= |arr1|
    invariant s1 == set(arr1[0..i])
{
    s1 := s1 + {arr1[i]};
    i := i + 1;
}

var s2: set<int> := {};
var j := 0;

```

```

while j < |arr2|
  invariant 0 <= j <= |arr2|
  invariant s2 == set(arr2[0..j])
{
  s2 := s2 + {arr2[j]};
  j := j + 1;
}

areSameSet := (s1 == s2);
}

```

## Program → GenAI

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

```

problem_data/p6_10_same_set/program.dfy(14,23): Error: invalid Ident
  |
14 | invariant s1 == set(arr1[0..i])
  | ^

problem_data/p6_10_same_set/program.dfy(15,2): Error: invalid Suffix
  |
15 | {
  | ^

problem_data/p6_10_same_set/program.dfy(14,23): Error: missing semicolon at end of
  ↪ statement
  |
14 | invariant s1 == set(arr1[0..i])
  | ~~~~~

problem_data/p6_10_same_set/program.dfy(24,23): Error: invalid Ident
  |
24 | invariant s2 == set(arr2[0..j])
  | ^

problem_data/p6_10_same_set/program.dfy(25,2): Error: invalid Suffix
  |
25 | {
  | ^

problem_data/p6_10_same_set/program.dfy(24,23): Error: missing semicolon at end of
  ↪ statement
  |
24 | invariant s2 == set(arr2[0..j])
  | ~~~~~

6 parse errors detected in program.dfy

```

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-Co5BxEkaJdL1WJipBXwCEsvD4ykUz

**Model:** gpt-5.1-chat-latest

**Created at:** 1766052393

**Finish reason:** stop

**Usage:** 1411 tokens in, and 219 tokens out

```
{  
    var s1: set<int> := {};  
    var i := 0;  
    while i < |arr1|  
        invariant 0 <= i <= |arr1|  
        invariant s1 == set k:int | 0 <= k < i :: arr1[k]  
    {  
        s1 := s1 + {arr1[i]};  
        i := i + 1;  
    }  
  
    var s2: set<int> := {};  
    var j := 0;  
    while j < |arr2|  
        invariant 0 <= j <= |arr2|  
        invariant s2 == set k:int | 0 <= k < j :: arr2[k]  
    {  
        s2 := s2 + {arr2[j]};  
        j := j + 1;  
    }  
  
    areSameSet := (s1 == s2);  
}
```

## Program → GenAI

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

```
problem_data/p6_10_same_set/program.dfy(9,0): Error: a postcondition could not be proved  
    ↪ on this return path  
    |  
9 | {  
    | ^  
  
problem_data/p6_10_same_set/program.dfy(8,20): Related location: this is the  
    ↪ postcondition that could not be proved  
    |  
8 | ensures areSameSet == (array_to_set(arr1) == array_to_set(arr2))  
    | ^^
```

```
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-Co5C2u5ASx9cZ0xPtodefqeRZc3qN

**Model:** gpt-5.1-chat-latest

**Created at:** 1766052398

**Finish reason:** stop

**Usage:** 1663 tokens in, and 531 tokens out

```
{
    var s1: set<int> := {};
    var i := |arr1|;
    while i > 0
        invariant 0 <= i <= |arr1|
        invariant s1 == array_to_set(arr1[i..])
    {
        s1 := s1 + {arr1[i-1]};
        i := i - 1;
    }

    var s2: set<int> := {};
    var j := |arr2|;
    while j > 0
        invariant 0 <= j <= |arr2|
        invariant s2 == array_to_set(arr2[j..])
    {
        s2 := s2 + {arr2[j-1]};
        j := j - 1;
    }

    areSameSet := (s1 == s2);
}
```

## Final Program

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

method p6_10_same_set(arr1: seq<int>, arr2: seq<int>) returns (areSameSet: bool)
    ensures areSameSet == (array_to_set(arr1) == array_to_set(arr2))
```

```

{
    var s1: set<int> := {};
    var i := |arr1|;
    while i > 0
        invariant 0 <= i <= |arr1|
        invariant s1 == array_to_set(arr1[i..])
    {
        s1 := s1 + {arr1[i-1]};
        i := i - 1;
    }

    var s2: set<int> := {};
    var j := |arr2|;
    while j > 0
        invariant 0 <= j <= |arr2|
        invariant s2 == array_to_set(arr2[j..])
    {
        s2 := s2 + {arr2[j-1]};
        j := j - 1;
    }

    areSameSet := (s1 == s2);
}

```

## Total Token Usage

**Input tokens:** 4175  
**Output tokens:** 1354  
**Reasoning tokens:** 512  
**Sum of ‘total tokens’:** 5529

## Experiment Timings

Overall Experiment started at 1766052385892, ended at 1766052407800, lasting 21908ms (21.91 seconds)  
**Iteration #4** started at 1766052399903, ended at 1766052407800, lasting 7897ms (7.90 seconds)  
**Iteration #1** started at 1766052385892, ended at 1766052390478, lasting 4586ms (4.59 seconds)  
**Iteration #2** started at 1766052390478, ended at 1766052395341, lasting 4863ms (4.86 seconds)  
**Iteration #3** started at 1766052395341, ended at 1766052399903, lasting 4562ms (4.56 seconds)