

## Homework 9

Please submit your solution via email to the instructor with CC to ly603@nyu.edu.

The deadline for Homework 9 is April 15.

### Problem 1 Universes Type System (10 Points)

On the course web site you find an implementation of a heap. The heap is realized by an array, and every element in the heap stores its position in the heap. A method to enqueue new elements into the heap, and a method to remove an element from the heap are provided. The Universes Type System is used to guard the invariant that every element knows its current position. Unfortunately, this does not work. Explain the problem(s) and discuss why they cannot be solved with this type system.

### Problem 2 Maps with Binary Search Trees (15 Points)

Implement and verify a map ADT based on binary search trees in Dafny. Both the type of the domain of the map ADT and the type of the codomain should be `int`. Your data structure should provide the following operations:

- A constructor: **constructor** `Init()`  
that constructs an empty map.
- A method: **method** `inDomain(k: int)` **returns** `(present: bool)`  
that checks whether there exists a mapping for key `k` in the map.
- A method: **method** `Insert(k: int, v: int)` **returns** `(oldValue: int)`  
that inserts a key/value pair into the map. If a mapping for that key already exists, the old mapping will be replaced. The method returns the new value `v` if the key was not mapped to a value before this method has been called, and the old value otherwise.

Provide method contracts that specify the intended behavior and use Dafny to check that your implementation respects all contracts. Dafny should not report any errors.

*Hint:* See the `TreeSet` example from Lecture 9 to get started. Use three ghost fields to specify your operations: a field `Repr` for the dynamic frames, a field `Domain` for the domain of the map, and a field `Contents` for the actual map representation:

```
class TreeMap {
  ghost var Repr: set<object>;
  ghost var Domain: set<int>;
  ghost var Contents: map<int, int>;
  ...
}
```