Answer

The problem has a solution only if the gcd of the set of jug capacities divides *Goal*. The set of jug capacities is written in TLA$^+$ as $\{Capacity[j] : j \in Jugs\}$. Therefore, in terms of the definitions from the *GCD* module, the problem has a solution only if

$$Divides(SetGCD(\{Capacity[j] : j \in Jugs\}),\ Goal)$$

is true. That this is a necessary condition follows from the fact that the algorithm maintains the following invariant: the gcd of the set of jug capacities divides the amount of water in each jug. This invariant is written in TLA$^+$ as

$$\forall\, j \in Jugs : Divides(SetGCD(\{Capacity[k] : k \in Jugs\}),\ injug[j])$$

Modify the *DieHarder* spec so it imports the *GCD* module, and have TLC check that this is indeed an invariant. (Unless you put the *GCD* spec in a library folder, the file `GCD.tla` [or a copy of it] has to be in the same folder as the *DieHarder* spec.) Can you prove that the formula above is an invariant of algorithm *DieHarder*?

The necessary and sufficient condition for the existence of a solution depends on what it means for the heroes to "obtain" *Goal* gallons of water. If we require that those *Goal* gallons must be in the jugs, then the jugs obviously must have the capacity to hold that much water. This together with the requirement that the gcd of the jug capacities divides *Goal* implies that there does exist a solution. You may be able to find a proof of this on the Web, but it's more fun trying to prove it yourself. The proof I devised is based on the number-theoretic result of Question 4.6$^{\square}$.

CLOSE