Algorithm *DieHarder*

```
──────────────────── MODULE DieHarder ────────────────────
EXTENDS Integers

Min(m, n) ≜ IF m < n THEN m ELSE n
──────────────────────────────────────────────────────────
CONSTANTS Goal, Jugs, Capacity

ASSUME ∧ Goal ∈ Nat
       ∧ Capacity ∈ [Jugs → Nat \ {0}]

(********************************************************
--algorithm DieHarder {
  variable injug = [j ∈ Jugs ↦ 0] ;
  { while ( TRUE )
     { either with ( j ∈ Jugs )   fill jug j
                { injug[j] := Capacity[j] }
       or     with ( j ∈ Jugs )   empty jug j
                { injug[j] := 0 }
       or     with ( j ∈ Jugs, k ∈ Jugs \ {j} )   pour from jug j to jug k
                { with ( poured =
                            Min(injug[j] + injug[k], Capacity[k]) − injug[k] )
                    { injug[j] := injug[j] − poured ||
                      injug[k] := injug[k] + poured
                    }
                }
     }
   }
 }
********************************************************)
──────────────────────────────────────────────────────────
```

CLOSE