Example: Quicksort

```
{- lo >= 0 && hi >= 0 -}
{- lo < hi -}
{- length(A) > 0 -}
algorithm quicksort(A:List Int, lo:Int, hi:Int) {
  p := partition(A, lo, hi)
  quicksort(A, lo, p)
  quicksort(A, p + 1, hi)
}
```

```
algorithm partition(A:List Int, lo:Int, hi:Int) {
    mid = (lo + hi) / 2
    pivot := A[mid]
    …
}
```

21

Example: Quicksort

```
{- lo >= 0 && hi >= 0 -}  ✗
{- lo < hi -}             ✗
{- length(A) > 0 -}       ✗
algorithm quicksort(A:List Int, lo:Int, hi:Int) {
   p := partition(A, lo, hi)
   quicksort(A, lo, p)
   quicksort(A, p + 1, hi)
}
```

```
algorithm partition(A:List Int, lo:Int, hi:Int) {
    mid = (lo + hi) / 2
    pivot := A[mid]
    …
}
```

Example: Quicksort

```
{- lo >= 0 && hi >= 0 -}
{- lo < hi -}
{- length(A) > 0 -}
algorithm quicksort(A:List Int, lo:Int, hi:Int) {
   if (lo >= 0 && hi >= 0 && lo < hi && length(A) > 0) {
     p := partition(A, lo, hi)
     quicksort(A, lo, p)
     quicksort(A, p + 1, hi)
   } else {
      return ERROR
   }
}
```

```
algorithm partition(A:List Int, lo:Int, hi:Int) {
   mid = (lo + hi) / 2
   pivot := A[mid]
   …
}
```