

```

find Min
pre cond: data is an array of
           ints of data.length items
int min = data[0];
int i = 1;
while (i < data.length) {
    if (data[i] < min)
        min = data[i];
    i++;
}
post cond: min ≤ data[j] for
           0 ≤ j < data.length

I(n): i = n + 1 and
      min ≤ data[k] for
           0 ≤ k ≤ n
    
```

Oct 22-10:02 AM

```

Basis: I(0): (a) i = 1 and
             (b) min = data[j]
             (a) is true from line 2's execution
             (b) min = data[0] from line 2
Induction: if G ∧ I(k) then
            I(k+1)
Suppose: G ∧ I(k) are true
so i < data.length and
min = data[j] 0 ≤ j ≤ k
and i = k + 1
case 1: data[k+1] < min
        so min is now data[k+1]
case 2: data[k+1] ≥ min
        so min = data[k+1]
From ind. hyp. min ≤ data[j]
           0 ≤ j ≤ k and min ≤ data[k+1]
so min ≤ data[j] 0 ≤ j ≤ k+1
And i was k+1 and i++ adds 1
to i so i = k+2
    
```

Oct 22-10:15 AM

```

Eventual falsity of the guard
after (data.length) iterations
i = data.length from invariant
I(data.length - 1)
causing the guard to be
false.

Correctness of the post cond.
min = data[j] for
0 ≤ j < data.length
so I(data.length - 1) is true after
the guard fails
so min = data[k]
for 0 ≤ k < data.length
(k < data.length)
    
```

Oct 22-10:31 AM