22 Oct 2022

```
i = 0
*while (i ≤ 10)
    code
    code
    code
end while
```

The loop guard "the bouncer of the loop"

```
for (int i = 0; i ≤ 10; i++)
    code
    code
    code
end for
```

equivalent

```
i = 0.
*while (i ≤ 10)
    code
    code
    code
    i++
end while
```

1st time: i was set to 0

any other time: i incremented before asking the loop guard

```
for i = 1, 2, ..., 10
    code
    code
    code
end for
```

can be expanded as

can also think of

Loop Guard: $i \in \{1, 2, \ldots, 10\}$

Loop Invariants:

The statements: L = loop invariant

P = pre conditions

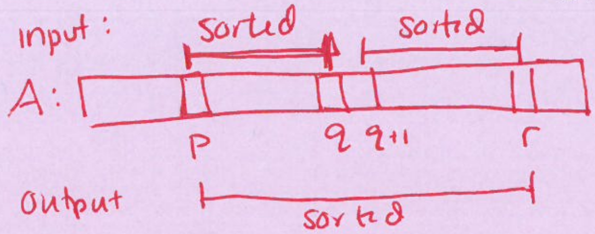
Q = post conditions

G = loop guard

The three things to prove:

1. INITIALIZATION: $P \Rightarrow L$
2. MAINT: $L_i \wedge G \Rightarrow L_{i+1}$, where L_i = the instant I am asking the loop guard for the i^{th} time if I can enter the array.
3. END: $L \wedge \neg G \Rightarrow Q$

22 Oct 2022




```
MERGE (A, p, q, r)
n ← q-p+1
m ← r-q
L ← new array, length n+1, L[n+1] = ∞
R ← new array, length m+1, R[m+1] = ∞
for (int i=1; i ≤ n; i++)
|   L[i] ← A[p+i-1]
end for
for j=1 to m
|   R[j] ← A[q+j]
end for
i ← 1
j ← 1
k ← p
*while k ≤ p
|   if L[i] ≤ R[j]
|   |   A[k] ← L[i]
|   |   i++
|   else
|   |   A[k] ← R[j]
|   |   j++
|   endif
endwhile
```

for loop in lines 5 – 7:

④ $L =$ "$\underbrace{\text{L is an array of length } n+1}_{l_1}$ and $\underbrace{L[1 \dots i-1] = A[p \dots p+i-2]}_{l_2 = l_2(i)}$ and $\underbrace{L[n+1] = \infty}_{l_3}$ and $\underbrace{i \in \mathbb{Z}}_{l_4 = l_4(i)}$ and $\underbrace{i \le n+1}_{l_5 = l_5(i)}$"

③ $P =$ "$i = 1$ and $n = q-p+1$ and $A[p \dots q]$ is sorted and L is an array of length $n+1$ and $L[n+1] = \infty$"

② $Q =$ "$\underbrace{\text{L is an array of length } n+1}_{q_1}$ such that $\underbrace{L[1 \dots n] = A[p \dots p+n-1]}_{q_2}$ and $\underbrace{L[n+1] = \infty}_{q_3}$"

① $G =$ "$i \le n$"

$\neg G =$ "$i > n$"

Proof

INITIALIZATION: $P \Rightarrow L$

Assume P. That is, assume that $i = 1$, $n = q-p+1$, L is an array of length $n+1$, $A[p \dots q]$ is sorted, $L[n+1] = \infty$

By assumption, l_1 and l_3 are true.

Since $i = 1$, l_4 and l_5 also hold.

Since $i = 1$, then $L[1 \dots i-1] = L[1 \dots 1-1] = \emptyset$ array and $A[p \dots p+i-2] = A[p \dots p-2] = \emptyset$ array.

So, vacuously, $L[1 \dots i-1] = A[p \dots p+i-1]$ ⊞

END: $L \wedge \neg G \Rightarrow Q$

Assume $\neg G$ and L. That is, assume $i > n$ and l_1 and l_2 and l_3

By assumption, (l_1) L is an array of length $n+1$ and (l_3) $L[n+1] = \infty$, so we just need to show $L[1 \dots n] = A[p \dots p+n-1]$.

By G ($i > n$) and (l_5) $i \le n+1$ and (l_4) $i \in \mathbb{Z}$, we know $i = n+1$.

Since $i = n+1$ and by l_2, we have $L[1 \dots n] = A[p \dots p+n-1]$, which is exactly q_2. ⊞

MAINT: $L_i \wedge G \Rightarrow L_{i+1}$

Assume L_i and G. (Expand out what that means).

In Line 6, $L[i]$ was set to $A[p+i-1]$.

Since we already had $L[1 \ldots i-1] = A[p - i-2]$, by $\ell_2(i)$ from L_i, we now have $L[1 \ldots i] = A[p \ldots p+i-1]$, which is $\ell_2(i+1)$.

Note: Don't forget ℓ_1, ℓ_3, $\ell_4(i+1)$, and $\ell_5(i+1)$! ∎