

Tutorials - Week 10 - Nov 14

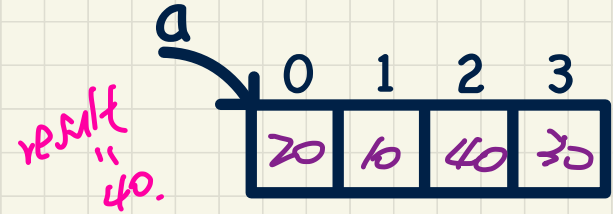
Graphs

***Precondition and Postcondition
Deriving and Tracing Loop Invariant
Correctness of Loops***

Iterative Algorithm: Precondition and Postcondition

```
1 int findMax (int[] a) {  
2   assert 0; /* precondition satisfied? */  
3   int i = 0;  
4   int result = a[i];  
5   assert LI; /* invariant established? */  
6   while(i != a.length) {  
7     if(a[i] > result) {  
8       result = a[i];  
9     }  
10    i = i + 1;  
11    assert LI; /* invariant preserved? */  
12  }  
13  assert R; /* postcondition satisfied? */  
14  return result;  
15 }
```

result is the temporary computed ^{max}value so far; index i ($\approx$ D value)



Q. $a \neq \text{null} \wedge$
 $a.length > 0$
or $\neq$

R. $\text{result} \geq \text{each value in } a$

$\text{result} \geq a[0]$

$\text{result} \geq a[1]$

$\vdots$

$\text{result} \geq a[3]$

$\vdots$

$\text{result} \geq a[n-1]$

$\vdots$

in Dijkstra's algo.

the loop of array already considered.

a

0	1	2	3
20	10	40	30

result $\geq$ each value in a

$$\forall j. 0 \leq j \leq \text{a.length} - 1 \Rightarrow$$

$< \text{a.length}$

largest index
(right-most)

if the j value is within this range

$$\frac{a[j] \leq \text{result}}{\checkmark \text{result} \geq a[j]}$$

$\hookrightarrow$ then this condition should be true.

$\hookrightarrow$ to derive the LI, result may only be applicable as some part/prefix of the array a.

Q (precondition) ~ input

R (postcondition)

~ input

~ output

~ no need to mention local variables

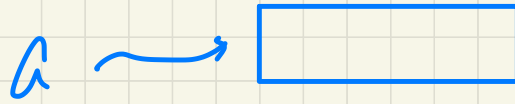
LI (Loop invariant)

~ input

~ output

~ loop counter

~ other local variable(s)



$a \neq \text{null}$

`int[] a = new int[0];`



$a \neq \text{null} \text{ \& \& } a.\text{length} == 0$

Iterative Algorithm: Loop Invariant (1)

```

1  int findMax (int[] a) {
2      assert Q; /* precondition satisfied? */
3      int i = 0;
4      int result = a[i];
5      assert LI; /* invariant established? */
6      while (i != a.length) {
7          if (a[i] > result) {
8              result = a[i];
9          }
10         i = i + 1;
11         assert LI; /* invariant preserved? */
12     }
13     assert R; /* postcondition satisfied? */
14     return result;
15 }

```



Proposed LI: *inappropriate!*
 $\forall j. 0 \leq j \leq i \Rightarrow \text{result} \geq a[j]$

after it.	i	result	LI	B
Start	0	20	✓ est.	✓
1	1	20	* i=1 ✓	✓
2	2	20	* i=2 ✗	

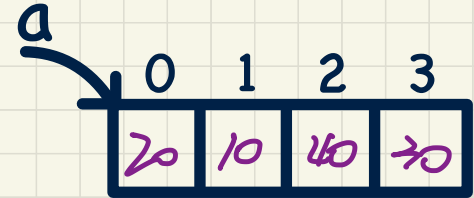
LI violated
 $20 \not\geq a[2]$
 "falsch"

* $\forall j. 0 \leq j \leq 1 \Rightarrow \text{result} \geq a[j]$
 ** $\forall j. 0 \leq j \leq 2 \Rightarrow \text{result} \geq a[j]$

Iterative Algorithm: Loop Invariant (2)

false $\Rightarrow P \equiv \text{true}$

```
1 int findMax (int[] a) {
2   assert Q; /* precondition satisfied? */
3   int i = 0;
4   int result = a[i];
5   assert LI; /* invariant established? */
6   while(i != a.length) {
7     if(a[i] > result) {
8       result = a[i];
9     }
10    i = i + 1;
11    assert LI; /* invariant preserved? */
12  }
13  assert R; /* postcondition satisfied? */
14  return result;
15 }
```



Fixed LI: $\leq i-1$

$\forall j. 0 \leq j < i \Rightarrow \text{result} \geq a[j]$

Exercise

show this LI is appropriate:

(1) established

(2) maintained.

$\forall j. 0 \leq j < i \Rightarrow \text{result} \geq a[j]$
 $\hookrightarrow \text{false}$

Iterative Algorithm: Correctness

```
1 int findMax (int[] a) {  
2   assert Q; /* precondition satisfied? */  
3   int i = 0;  
4   int result = a[i];  
5   assert LI; /* invariant established? */  
6   while (i != a.length) {  
7     if (a[i] > result) {  
8       result = a[i];  
9     }  
10    i = i + 1;  
11    assert LI; /* invariant preserved? */  
12  }  
13  assert R; /* postcondition satisfied? */  
14  return result;  
15 }
```

Assuming the antecedent

③ Substitute i by $a.length$ in A_2 : this gives us the consequent to prove. ✓

$\neg B \wedge LI \Rightarrow R$
exit from loop maintained in last iteration. post-condition

A_1
 $\neg (i \neq a.length)$
 $\wedge A_2$
 $\forall j. 0 \leq j < i \Rightarrow result \geq a[j]$
 $\Rightarrow \forall j. 0 \leq j < a.length \Rightarrow result \geq a[j]$