

# Termination

1

## BUBBLESORT: Partial Correctness

If BUBBLESORT terminates, then it returns a sorted array.

```

@pre  $\top$ 
@post sorted(rv, 0, |a| - 1)
int [] BUBBLESORT(int [] a0) {
  int [] a := a0;
  for (int i := |a| - 1; i > 0; i := i - 1) {
    for (int j := 0; j < i; j := j + 1) {
      if (a[j] > a[j + 1]) {
        int t := a[j];
        a[j] := a[j + 1];
        a[j + 1] := t;
      }
    }
  }
  return a;
}

```

2

## BUBBLESORT: Termination

Does BUBBLESORT always terminate?

```

@pre  $\top$ 
int [] BUBBLESORT(int [] a0) {
  int [] a := a0;
  for (int i := |a| - 1; i > 0; i := i - 1) {
    for (int j := 0; j < i; j := j + 1) {
      if (a[j] > a[j + 1]) {
        int t := a[j];
        a[j] := a[j + 1];
        a[j + 1] := t;
      }
    }
  }
  return a;
}

```

3

## BUBBLESORT: Termination

Well-founded  $(\underbrace{\mathbb{Z}^+ \times \mathbb{Z}^+}_S, \underbrace{<}_<)$

```

@pre  $\top$ 
int [] BUBBLESORT(int [] a0) {
  int [] a := a0;
  for @L1 :  $\underbrace{(i + 1 \geq 0)}_{\mu_1} \downarrow \underbrace{(i + 1, i + 1)}_{\delta_1}$ 
    (int i := |a| - 1; i > 0; i := i - 1)
    for @L2 :  $\underbrace{(i - j \geq 0 \wedge i + 1 \geq 0)}_{\mu_2} \downarrow \underbrace{(i + 1, i - j)}_{\delta_2}$ 
      (int j := 0; j < i; j := j + 1) {
        if (a[j] > a[j + 1]) {
          int t := a[j];
          a[j] := a[j + 1];
          a[j + 1] := t;
        }
      }
  }
  return a;
}

```

$\mu_i$ : supporting invariant  
 $\delta_i$ : ranking function

4

### BUBBLESORT: Supporting Invariants

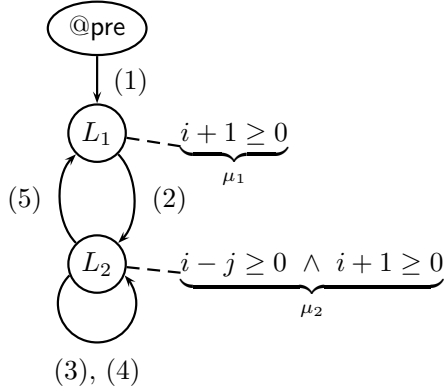
Path (5)

$@L_2 : i - j \geq 0 \wedge i + 1 \geq 0$

assume  $j \geq i$ ;

$i := i - 1$ ;

$@L_1 : i + 1 \geq 0$



$$\underbrace{i - j \geq 0 \wedge i + 1 \geq 0}_{\mu_2} \wedge j \geq i \Rightarrow \underbrace{(i - 1) + 1 \geq 0}_{\mu'_1}$$

5

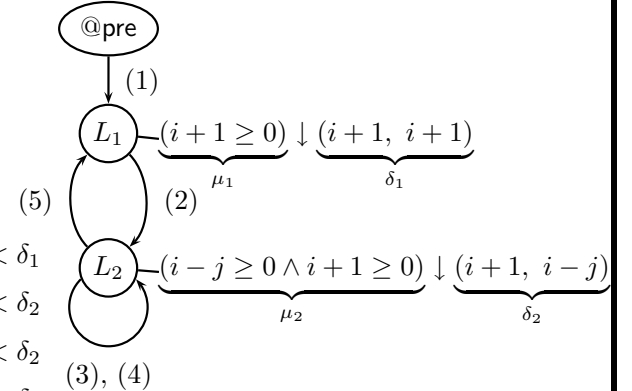
### BUBBLESORT: Ranking Functions

Bounded below:

- $\mu_1 \Rightarrow \delta_1 \geq 0$
- $\mu_2 \Rightarrow \delta_2 \geq 0$

Decreasing:

- $\mu_1 \wedge \pi_2 \Rightarrow \delta'_2 < \delta_1$
- $\mu_2 \wedge \pi_3 \Rightarrow \delta'_2 < \delta_2$
- $\mu_2 \wedge \pi_4 \Rightarrow \delta'_2 < \delta_2$
- $\mu_2 \wedge \pi_5 \Rightarrow \delta'_1 < \delta_2$



6

### BUBBLESORT: Bounded Below

$$\bullet \underbrace{i + 1 \geq 0}_{\mu_1} \Rightarrow \underbrace{(i + 1, i + 1) \geq 0}_{\delta_1}$$

$$\bullet \underbrace{i - j \geq 0 \wedge i + 1 \geq 0}_{\mu_2} \Rightarrow \underbrace{(i + 1, i - j) \geq 0}_{\delta_2}$$

7

### BUBBLESORT: Decreasing

Path 2

$@L_1 : (i + 1 \geq 0) \downarrow (i + 1, i + 1)$

assume  $i > 0$ ;

$j := 0$ ;

$@L_2 : (i - j \geq 0 \wedge i + 1 \geq 0) \downarrow (i + 1, i - j)$

$$\underbrace{i + 1 \geq 0}_{\mu_1} \wedge i > 0 \Rightarrow \underbrace{(i + 1, i - 0)}_{\delta'_2} < \underbrace{(i + 1, i + 1)}_{\delta_1}$$

8

### BUBBLESORT: Decreasing

#### Path 3/4

@L<sub>2</sub> :  $(i - j \geq 0 \wedge i + 1 \geq 0) \downarrow (i + 1, i - j)$

assume  $j < i$ ;

:

$j := j + 1$ ;

@L<sub>2</sub> :  $(i - j \geq 0 \wedge i + 1 \geq 0) \downarrow (i + 1, i - j)$

$$\underbrace{i - j \geq 0 \wedge i + 1 \geq 0}_{\mu_2} \wedge j < i \Rightarrow$$
$$\underbrace{(i + 1, i - (j + 1))}_{\delta'_2} < \underbrace{(i + 1, i - j)}_{\delta_2}$$

9

### BUBBLESORT: Decreasing

#### Path 5

@L<sub>2</sub> :  $(i - j \geq 0 \wedge i + 1 \geq 0) \downarrow (i + 1, i - j)$

assume  $j \geq i$ ;

$i := i - 1$ ;

@L<sub>1</sub> :  $(i + 1 \geq 0) \downarrow (i + 1, i + 1)$

$$\underbrace{i - j \geq 0 \wedge i + 1 \geq 0}_{\mu_2} \wedge j \geq i \Rightarrow$$
$$\underbrace{((i - 1) + 1, (i - 1) + 1)}_{\delta'_1} < \underbrace{(i + 1, i - j)}_{\delta_2}$$

10

### BINARYSEARCH: Partial Correctness

If array  $a$  is sorted and BINARYSEARCH terminates, then it returns  $\top$  iff it contains  $e$  in positions  $[\ell, u]$ .

@pre  $0 \leq \ell \wedge u < |a| \wedge \text{sorted}(a, 0, |a| - 1)$

@post  $rv \leftrightarrow \exists i. \ell \leq i \leq u \wedge a[i] = e$

```
bool BSEARCH(int[] a, int l, int u, int e) {
  if (l > u) return false;
  else {
    int m := (l + u) div 2;
    if (a[m] = e) return true;
    else if (a[m] < e) return BSEARCH(a, m + 1, u, e);
    else return BSEARCH(a, l, m - 1, e);
  }
}
```

11

### BINARYSEARCH: Termination

Does BINARYSEARCH terminate for  $\ell, u$ , and  $a$  satisfying the precondition?

@pre  $0 \leq \ell \wedge u < |a|$

```
bool BSEARCH(int[] a, int l, int u, int e) {
  if (l > u) return false;
  else {
    int m := (l + u) div 2;
    if (a[m] = e) return true;
    else if (a[m] < e) return BSEARCH(a, m + 1, u, e);
    else return BSEARCH(a, l, m - 1, e);
  }
}
```

12

### BINARYSEARCH: Termination

Well-founded ( $\mathbb{Z}^+$ ,  $<$ )

@pre  $u - \ell + 1 \geq 0$

@post  $\top$

$\downarrow (u - \ell + 1)$

```
bool BSEARCH(int[] a, int  $\ell$ , int  $u$ , int e) {  
  if ( $\ell > u$ ) return false;  
  else {  
    int  $m := (\ell + u) \text{ div } 2$ ;  
    if ( $a[m] = e$ ) return true;  
    else if ( $a[m] < e$ ) return BSEARCH(a,  $m + 1, u, e$ );  
    else return BSEARCH(a,  $\ell, m - 1, e$ );  
  }  
}
```

13

### BINARYSEARCH: Termination

$$\ell \leq u \Rightarrow u - \underbrace{((\ell + u) \text{ div } 2) + 1}_{m+1} + 1 < u - \ell + 1$$

$$\begin{aligned} \ell &= (\ell + \ell) \text{ div } 2 \\ &\leq (\ell + u) \text{ div } 2 \quad \text{since } \ell \leq u \\ &< (\ell + u) \text{ div } 2 + 1 \end{aligned}$$

14

### BINARYSEARCH: Termination

$$\ell \leq u \Rightarrow \underbrace{(((\ell + u) \text{ div } 2) - 1)}_{m-1} - \ell + 1 < u - \ell + 1$$

$$\begin{aligned} u &= (u + u) \text{ div } 2 \\ &\geq (\ell + u) \text{ div } 2 \quad \text{since } \ell \leq u \\ &> (\ell + u) \text{ div } 2 - 1 \end{aligned}$$

15

### BINARYSEARCH: Termination

Path 1:

@pre  $(u - \ell + 1 \geq 0) \downarrow (u - \ell + 1)$

assume  $\ell \leq u$ ;

$m := (\ell + u) \text{ div } 2$ ;

assume  $a[m] \neq e$ ;

assume  $a[m] < e$ ;

@pre  $(u - \ell + 1 \geq 0) \{ \ell \mapsto m + 1 \} \downarrow (u - \ell + 1) \{ \ell \mapsto m + 1 \}$

$$\ell \leq u \Rightarrow u - ((\ell + u) \text{ div } 2) + 1 < u - \ell + 1$$

16