

Problem 1

1. Loop Invariant: $y \geq 0 \wedge result + xy = mn$

2. Base Case: $result = 0 \wedge x = m \wedge y = n \implies result + xy = mn$.

$y = 0 \implies y \geq 0$

3. Assume $result + xy = mn$ holds at k iteration, and prove $result + xy = mn$ holds at $k + 1$ iteration. Consider two exhaustive cases:

a. If y is even:

$x_1 = x + x \wedge y_1 = y/2 \implies y_1 \geq 0 \wedge result + xy = mn \implies result_1 + xy = result + (x + x)(y/2) = result + xy = mn$

Thus the Inductive hypothesis holds at the case when y is even.

b. Else(The odd case):

$result_1 = result + x \wedge y_1 = y - 1 \geq 0 \implies result_1 + xy_1 = result + x + xy_1$
 $= result + x + x(y - 1) = result + x + xy - x = result + xy = mn$

Thus the Inductive hypothesis holds at the case when y is odd.

4. At exit: $!(y = 0) \wedge result + xy = mn$

$\implies y = 0 \wedge result + xy = mn$

$\implies result + 0 = mn$

$\implies result = mn$

5. $D = y$

When y is even: $y_1 = y/2 \implies y_1 < y$. Thus D decreases.

When y is odd: $y_1 = y - 1 \implies y_1 < y$. Thus D decreases.

Since $D = y$ decreases in both cases. D is decrementing in every iteration.

When D reaches its minimum 0, $y = 0$. When $y = 0$, the loop exits.

Problem2

a.

Input: A directed non-empty array with "blue" and "red"

Output: A sorted array with 0 $k - 1$ being red and k $N - 1$ being blue.

Algorithm 1 Dutchflag

```
1: procedure DUTCHFLAG(arr)
2:    $i = 0$ 
3:    $j = \text{len}(\text{arr}) - 1$ 
4:   while  $i \leq j$  do
5:     if  $\text{arr}[i] == \text{"blue"}$  and  $\text{arr}[j] == \text{"red"}$  then
6:        $\text{swap}(i, j, \text{arr})$ 
7:        $i++ = 1$ 
8:        $j-- = 1$ 
9:     if  $\text{arr}[i] == \text{"red"}$  and  $\text{arr}[j] == \text{"blue"}$  then
10:       $i++ = 1$ 
11:       $j-- = 1$ 
12:     if  $\text{arr}[i] == \text{"red"} \wedge \text{arr}[j] == \text{"red"}$  then
13:        $i++ = 1$ 
14:     if  $\text{arr}[i] == \text{"blue"} \wedge \text{arr}[j] == \text{"blue"}$  then
15:        $j-- = 1$ 
16:    $k = j + 1$ 
```

b.

$0 \leq k \leq \text{arr.Length}$

$\text{forall } e :: 0 \leq e < k \implies \text{arr}[e] == \text{"red"}$

$\text{forall } f :: k \leq f < \text{arr.Length} \implies \text{arr}[f] == \text{"blue"}$

c.

$\text{forall } d :: 0 \leq d \leq \text{arr.Length} - 1 \implies \text{arr}[d] == 'b' \vee \text{arr}[d] == 'r'$

$0 \leq i \leq \text{arr.Length}$

$-1 \leq j \leq \text{arr.Length} - 1$

$\text{forall } b :: 0 \leq b < i \implies \text{arr}[b] == 'r'$

$\text{forall } c :: j < c < \text{arr.Length} \implies \text{arr}[c] == 'b'$

Problem 3

Inner LI: $1 \leq s \leq r + 1 \wedge u == v * s$

Base Case:

$$r = 0, s = 1, u = 1, v = u \implies 1 \leq s \leq r + 1$$

In order to show $u == v * s$ hold at the base case, we substitute the values in and get $1 == 1 * 1$.

Inductive step: Assume the Inner LI holds for the base case, prove it holds at the $k + 1$ iteration.

$$r_k = 0$$

$$v_k = u_k$$

$$s_{k+1} = s_k + 1$$

$$u_{k+1} = u_k + v_k$$

$$1 \leq s_k \implies 1 \leq s_k + 1 \implies 1 \leq s_k + 1 \leq r_k + 1 \implies 1 \leq s_{k+1} \leq r_{k+1}$$

$$u_{k+1} = u_k + v_k \wedge u_k = v_k * s_k \implies v_{k+1} = v_k * (s_k + 1) = v_k(s_k) + v_k$$

$$u_k = u_{k+1} - v_k = v_k(s_k) + v_k - v_k$$

$$u_k = v_k * s_k$$

Thus the inductive steps hold.

Outer LI: $0 \leq r \leq n \wedge u == \text{Factorial}(r)$

Base Case:

$$u = 1 \wedge r = 0 \implies u = \text{Factorial}(r)$$

$$r = 0 \wedge n \geq 0 \implies 0 \leq r \leq n$$

Inductive step: Assume the Inner LI holds for the base case, prove it holds at the $k + 1$ iteration.

$$u_k + 1 = u_k * v_k$$

$$v_{new} = u_{old}$$

$$r_k + 1 = r_k + 1$$

$$0 \leq r_k + 1 \leq n \implies 0 \leq r_k + 1 \leq n$$

$$u_{new} = v * s = v * (r_{old} + 1) = \text{Factorial}(r_{old}) * r_{old} + 1 = \text{Factorial}(r_{old} + 1)$$

Thus the inductive steps hold.