Xinshi Wang
661975305
HW01

Problem 1

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

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 minumum 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

procedure DUTCHFLAG(arr)
    $i = 0$
    $k = len(arr) - 1$
    while $i < k$ **do**
        if $arr[i] == "red"$ **then**
            $i+ = 1$
        if $arr[k] == "blue"$ **then**
            $k- = 1$
        if $arr[i] == "blue" \wedge arr[k] == "red"$ **then**
            $swap(arr, i, k)$

b.

forall $x < k, arr[x] == "red"$

forall $k - 1 < x < len(arr), arr[x] == "blue"$

c.

$i < k$

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 substitue 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: $0 \leq r \leq n \wedge u == Factorial(r)$

Base Case:

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

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

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

$u_k + 1 = u_k * v_k$

$v_{new} = u_{old}$

$r_k + 1 = r_k + 1$

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

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

Thus the inductive steps hold.