Kaleb Kim
9/20/24

CSDS 310 Assignment 1

Note: Arrays are zero-indexed.

Problem 1

a) **Loop Invariant**: At the start of each iteration, the greatest common divisor (GCD) of x and y is the same as the GCD of the original input integers a and b. In other words, $\gcd(x, y) = \gcd(a, b)$

b) **Initialization:** Before the first iteration, $x = a$ and $y = b$. Since these values are the same, $\gcd(x, y) = \gcd(a, b)$.

- **Maintenance**: At the start of the first iteration, the statement $\gcd(x, y) = \gcd(a, b)$ is true. In the loop, there are two cases:

  Case 1: $x > y$. In this case, $x = x - y$. Subtracting the smaller number, y, does not change the GCD. For proof, let $c = \gcd(x - y, y)$, in which $c \in \mathbb{Z}$ and $x > y$. By definition, $c(x - y) = cy$. Distributing, $cx - cy = cy$. Solving, $cx = 2cy = c(2y)$. As both sides of the equation remain integers, $c = \gcd(x, 2y) = \gcd(x - y, y) = \gcd(x, y)$.

  Case 2: $x \leq y$. In this case, $y = y - x$. This uses the same proof as Case 1, in which $\gcd(x, y) = \gcd(x, y - x)$.

  As the gcd remains same, the loop invariant holds.

- **Termination**: The loop terminates when $x = y$. By the loop invariant, $\gcd(a, b) = \gcd(x, y)$.

c) Inputs a and b are positive integers and $x = a$ and $y = b$, so x and y must be positive. In the loop, x and y decrease by the smaller counterpart. Given that x and y are positive, in both cases of subtraction, it is not possible for one number to subtract the other and be negative. The integers may not decrease below 1 either, since that case implies that the integers are equal, which is the termination condition.

d) Our loop invariant terminates when $x = y$, thus $x = x$. At this point in the algorithm, we have $\gcd(x, y) = \gcd(x, x)$. $x = x$, thus $x \mid x$, so we can conclude that $x = \gcd(x, x) = \gcd(x, y) = \gcd(a, b)$.

Problem 2

a)

```
procedure FINDPAIR(A, B, x):
    i ⟵ 0
    j ⟵ n - 1
    while i < n and j >= 0:
        if A[i] + B[j] == x then
            return i, j
        else if A[i] + B[j] < x then
            i ⟵ i + 1
        else:
            j ⟵ j - 1
    return FALSE
```

b)

- **Loop Invariant**: At the start of each iteration, if $A[i] + B[j] = x$, then indices i, j exist in the subarrays $A[i...n-1]$ and $B[0...j]$.
- **Initialization**: Initially, $i = 0$ and $j = n - 1$ so the subarrays in the loop invariant are $A[0...n-1]$ and $B[0...n-1]$, which are over the entire arrays A and B. Thus, the loop invariant holds initially.
- **Maintenance**: At the start of each iteration, if $A[i] + B[j] = x$, then indices i, j exist in the subarrays $A[i...n-1]$ and $B[0...j]$, which are returned. Else, given that A, B are sorted in nondecreasing order:

$$A[i+1] \geq A[i]$$
$$A[i+1] + B[j] \geq A[i] + B[j]$$

and

$$B[j-1] \leq B[j]$$
$$A[i] + B[j-1] \leq A[i] + B[j]$$

This means that if $A[i] + B[j] < x$, we increment i by 1, maintaining the subarray A[i...n-1]. Otherwise, in the other case that $A[i] + B[j] > x$, we decrement j by 1, maintaining the subarray B[0...j].
- **Termination**: The loop terminates if $A[i] + B[j] = x$ exists, since it returns the existing indices of i, j in subarrays $A[i...n-1]$ and $B[0...j]$, satisfying the loop invariant. Else, when $i \geq n$ or $j < 0$, meaning x does not exist in either subarray $A[i...n-1]$ or $B[0...j]$, no valid pair is found and the algorithm returns FALSE.

Problem 3

```
procedure NATURALSELECTION(m, n):
    l ⟵ m
    p ⟵ n
    while l + p > 1:
        pick two animals
        if both Pisidians then
            p ← p - 2
            l ← l + 1
        else:
            l ← l - 1
    if p mod 2 = 1:
        return PISIDIAN
    else:
        return LYDIAN
```

- **Loop Invariant**: At the start of each iteration, the parity of initial number of Pisidians n is the same as the remaining number of is Pisidians p. In other words, $p \bmod 2 = n \bmod 2$.
- **Initialization**: Intially, $l = m$ and $p = n$. Substituting in the loop variant, $n \bmod 2 = n \bmod 2$. Thus, the loop invariant holds.

- **Maintenance**: At the start of each iteration, two animals are picked. There are three resulting cases:
  - Lydian (l) and Lydian (l): As $l \leftarrow l - 1$, p is unchanged. $p \bmod 2 = p \bmod 2$, so the parity of p remains unchanged.
  - Pisidian (p) and Pisidian (p): Both Pisidians kill each other, so $p \leftarrow p - 2$. This means:

$$(p - 2) \bmod 2 = p \bmod 2 - 2 \bmod 2 = p \bmod 2$$

  Thus, the parity of p remains unchanged.
  - Lydian (l) + Pisidian (p): $l \leftarrow l - 1$, similar to the first case. Thus, the parity of p also remains unchanged.

As the parity of p does not change in all cases, the loop invariant holds.

- **Termination**: The loop terminates when there is one individual left, or $l + p = 1$. By the loop invariant, as the parity of n equals the parity of p, $p \bmod 2 = 1$ means that $p = 1$, thus the last standing population is the Pisidians. Otherwise, the last standing popuation is the Lydians.