CSED332 Assignment 4

Gwon Minjae

Problem 1

Consider the following program to find the maximum value in an array. Write a Hoare logic proof (decorated program) to prove the given Hoare triple.

$\{0 < N\}$

```
int m = A[0];
int i = 1;
while (i < N){
    if (A[i] > m)
        m = A[i];
    else
        skip;
    i = i + 1;
}
```

$\{m = max(A[0], A[1], \dots, A[N-1])\}$

Proof.

$\{0 < N\}$

```
int m = A[0];
int i = 1;
```

$\{1 \leq i \leq N \land m = max(A[0])\}$

```
while (i < N){
```

$\{1 \leq i \leq N \land m = max(A[0]) \land i < N\} \implies$

$\{1 \leq i + 1 \leq N \land m = max(A[0])\}$

```
    if (A[i] > m)
        m = A[i];
    else
        skip;
```

$\{1 \leq i + 1 \leq N \land m = max(A[0], A[i])\}$

```
    i = i + 1;
```

$\{1 \leq i \leq N \land m = max(A[0], A[i])\}$

```
}
```

$\{1 \leq i < N \land m = max(A[0], A[i])\} \implies$

$\{m = max(A[0], A[1], \dots, A[N-1])\}$

Problem 2

Write a Hoare logic proof (decorated program) to show that the given Hoare triple holds and the program always terminates (hint: what is a ranking function?).

$\{x \geq 0 \land y > 0\}$

```
int r = x;
int q = 0;
while (y <= r) {
    r = r - y;
    q = q + 1;
}
```

$\{x = qy + r \land 0 \leq r < y\}$

Proof.

$\{x \geq 0 \land y > 0\}$

```
int r = x;
int q = 0;
```

$\{x \geq 0 \land y > 0 \land x = qy + r\}$

```
while (y <= r) {
```

$\{x \geq 0 \land y > 0 \land x = qy + r \land y \leq r\} \implies$

$\{x \geq 0 \land y > 0 \land x = (q+1)y + (r-y) \land y \leq r\}$

```
    r = r - y;
```

$\{x \geq 0 \land y > 0 \land x = (q+1)y + r \land 0 \leq r\}$

```
    q = q + 1;
```

$\{x \geq 0 \land y > 0 \land x = qy + r \land 0 \leq r\}$

```
}
```

$\{x \geq 0 \land y > 0 \land x = qy + r \land 0 \leq r \land r < y\} \implies$

$\{x = qy + r \land 0 \leq r < y\}$

Problem 3

Consider the following program for sorting an array. Write a Hoare logic proof to prove the given Hoare triple, where $sorted(a_1, a_2, \ldots, a_k)$ means $a_1 \leq a_2 \leq \cdots \leq a_k$.

$\{0 \leq N\}$

```
int i = 1;
while (i < N) {
    int j = i;
    while (j > 0 && A[j-1] > A[j]) {
        int t = A[j-1];
        A[j-1] = A[j];
        A[j] = t;
        j = j - 1;
    }
    i = i + 1;
}
```

$\{sorted(A[0], A[1], A[2], \ldots, A[N-1])\}$

Prerequisite.

$$
\begin{aligned}
S(a,b) &= \begin{cases} sorted(A[a], \cdots, A[b]), & \text{if } 0 \le a \le b < N \\ \{\}, & \text{otherwise.} \end{cases} \\
L(a,b) &= \begin{cases} \{A[a], \cdots A[b]\}, & \text{if } 0 \le a \le b < N \\ \{\}, & \text{otherwise.} \end{cases} \\
R &= \text{(WLOG) Remaining part of A}
\end{aligned}
$$

Proof.

$\{0 \le N\}$

```
int i = 1;
```

$\{0 \le N \wedge i \ge 1 \wedge A = sorted(A[0]) + \{A[1], \cdots, A[N-1]\}\} \Longrightarrow$

$\{0 \le N \wedge i \ge 1 \wedge A = sorted(A[0], \cdots, A[i-1]) + \{A[i], \cdots, A[N-1]\}\}$

```
while (i < N) {
```

$\{0 \le N \wedge i \ge 1 \wedge A = sorted(A[0], \cdots, A[i-1]) + \{A[i], \cdots, A[N-1]\} \wedge i < N\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i < N \wedge A = sorted(A[0], \cdots, A[i-1]) + \{A[i], \cdots, A[N-1]\}\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = sorted(A[0], \cdots, A[(i+1)-2]) + \{A[(i+1)-1], \cdots, A[N-1]\}\} \Longrightarrow$

```
int j = i;
```

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = sorted(A[0], \cdots, A[(i+1)-2]) + \{A[(i+1)-1], \cdots, A[N-1]\} \wedge j \le i\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, j-2) + L(j-1, j) + S(j+1, i-1) + R \wedge j \le i\}$

```
while (j > 0 && A[j-1] > A[j]) {
```

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, j-2) + L(j-1, j) + S(j+1, i-1) + R \wedge j \le i$
$\wedge j > 0 \wedge A[j-1] > A[j]\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, j-2) + L(j-1, j) + S(j+1, i-1) + R$
$\wedge 0 \le j-1 \le i \wedge A[j-1] > A[j]\}$

```
int t = A[j-1];
A[j-1] = A[j];
A[j] = t;
```

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, (j-1)-2) + L((j-1)-1, j-1) + S((j-1)+1, i-1) + R$
$\wedge 0 \le j-1 \le i \wedge A[j-1] \le A[j]\}$

```
j = j - 1;
```

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, j-2) + L(j-1, j) + S(j+1, i-1) + R$
$\wedge 0 \le j \le i \wedge A[j] \le A[j+1]\}$

```
}
```

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = S(0, j-2) + L(j-1, j) + S(j+1, i-1) + R \wedge 0 \le j \le i \wedge A[j] \le A[j+1]\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = sorted(A[0], \cdots, A[i]) + \{A[(i+1)], \cdots, A[N-1]\}\} \Longrightarrow$

$\{0 \le N \wedge 1 \le i+1 \le N \wedge A = sorted(A[0], \cdots, A[(i+1)-1]) + \{A[(i+1)], \cdots, A[N-1]\}\}$

```
i = i + 1;
```

$\{0 \le N \wedge 1 \le i \le N \wedge A = sorted(A[0], \cdots, A[i-1]) + \{A[i], A[i+1], \cdots, A[N-1]\}\}$

```
}
```

$$\{0 \leq N \wedge 1 \leq i \leq N \wedge A = \mathit{sorted}(A[0], \cdots, A[i-1]) + \{A[i], A[i+1], \cdots, A[N-1]\} \wedge i \geq N\} \Longrightarrow$$

$$\{0 \leq N \wedge i = N \wedge A = \mathit{sorted}(A[0], \cdots, A[i-1]) + \{A[i], A[i+1], \cdots, A[N-1]\}\} \Longrightarrow$$

$$\{0 \leq N \wedge i = N \wedge A = \mathit{sorted}(A[0], \cdots, A[N-1])\} \Longrightarrow$$

$$\{\mathit{sorted}(A[0], A[1], A[2], \ldots, A[N-1])\}$$