CSC 240 Winter 2024 Homework Assignment 8

My name and student number: Haoyun (Bill) Xi 1009992019
The list of people with whom I discussed this homework assignment: None.

Recall the algorithm SRT that sorts an array $A[1...n]$ into nondecreasing order when n is a power of 2 and that calls an auxiliary function AUX.

```
SRT(A[1..n], n):
1   if n > 1 then
2       SRT(A[1..n/2], n/2)
3       SRT(A[(n/2 + 1)..n], n/2)
4       AUX(A[1..n], n)
```

```
AUX(A[1..n], n):
1   if n > 2 then
2       for i ← 1 to n/4 do
3           swap the value of A[i + n/4] and A[i + n/2]
4       AUX(A[1..n/2], n/2)
5       AUX(A[(n/2 + 1)..n], n/2)
6       AUX(A[(n/4 + 1)..3n/4], n/2)
7   else if A[1] > A[2] then
8       swap the value of A[1] and A[2]
```

1. What happens when AUX$(A, 4)$ is called, where $A = [2, 1, 4, 3]$? Briefly explain your answer.

   Solution.
   When $A = [2, 1, 4, 3]$, AUX$(A, 4)$ will not modify the list.
   Since $n > 2$, line 2-3 will first execute from $i = 1$ to $i = \frac{n}{4} = 1$, to swap $A[1 + \frac{n}{4}] = A[2]$ and $A[1 + \frac{n}{2}] = A[3]$. This results $A^3 = [2, 4, 1, 3]$. (**SPECIAL NOTE:** if a list variable is A, we use A^i to denote the state of A **after** the i^{th} line is executed. A^0 denote the original input.)

   Line 4 will compare the first and second element of $A[1..2]$. Since they are already in non-decreasing order, nothing is changed.
   Line 5 will compare the first and second element of $A[3..4]$. Since they are already in non-decreasing order, nothing is changed.
   Line 6 will compare the first and second element of $A[2..3]$. Since $A[2] > A[3]$, the code will swap them back, making $A^6 = [2, 1, 4, 3]$ being the original A^0.

2. Give specifications that are satisfied by AUX and from which you can prove that SRT is a correct sorting algorithm when n is a power of 2.

   Solution.

Preconditions:
$A[1..n]$ is an array with elements from a totally ordered set.
$n = 2^i$ for some integer $i \geq 1$.
First half ($A[1..\frac{n}{2}]$) and second half ($A[(\frac{n}{2}+1)..n]$) of A are sorted in non-decreasing order.
Formally, for every $x, y \in [n/2]$, $x < y$ IMPLIES $A[x] \leq A[y]$ AND $A[\frac{n}{2}+x] \leq A[\frac{n}{2}+y]$
Postconditions:
If **AUX**$(A[1..n], n)$ is performed, when it halts, A is sorted in non-decreasing order and the multiset of elements in A is unchanged.

3. Prove that AUX satisfies the specifications in your answer to question 2.

Solution.

Let $\mathcal{P}_2 = \{n | n = 2^i \text{ for some } i \in \mathbb{N}^+\} \subseteq \mathbb{Z}^+$ denote the set of powers of 2.
Alternatively, we can see $\mathcal{P}_2 \subseteq \mathbb{Z}^+$ as the recursively defined structure where the base case is $1 \in \mathcal{P}_2$ and the constructor case is $2 \cdot n \in \mathcal{P}_2$ for all $n \in \mathcal{P}_2$.
For $n \in \mathcal{P}_2 \setminus \{1\}$. Let $P(n) =$"for all arrays $A[1..n]$ satisfying the preconditions, if AUX(A[1..n], n) is performed, then it eventually halts, at which time A is sorted in non-decreasing order and the multiset of elements in A is unchanged."

We can omit $n = 1$ because it doesn't satisfy our precondition that $n = 2^i$ for some integer $i \geq 1$.
We will prove by structural induction that $\forall n \in \mathcal{P}_2 \setminus \{1\}.P(n)$.

Proof.

Let $n \in \mathcal{P}_2 \setminus \{1\}$ be arbitrary.
Base case: assume $n = 2$.
Let $A[1..2]$ be an arbitrary array of length 2. A must equal to some $[a, b]$ where a, b are from a totally ordered set. Since in line 1, $n > 2$ is not true, when AUX$(A[1..2], 2)$ is called, line $2 - 6$ will be omitted.
When $a \leq b$, the condition on line 7 is false and the program halts where nothing is changed. However, $a \leq b$ would implies the list is already sorted (in non-decreasing order).
When $a > b$, the condition on line 7 is true and line 8 will trigger. Thus a, b are swapped. Now, $A[1] = b$ and $A[2] = a$. By $a > b$, the list is sorted and the program halts.
Hence we can see AUX$(A[1..2], 2)$ satisfy the postconditions.
Since A is arbitrary, we proved $P(2)$.

Inductive case: assume $n \geq 4$ and $P(n/2)$ is true.
Let $A[1..n]$ be an arbitrary array of length n where by the properties of powers of 2, n must has a factor of 4. Further assume that A satisfy the precondition, thus $A[1..n/2]$ and $A[(n/2+1)..n]$ are both sorted.
For simplicity, we will call the first half $F = A^0[1..n/2]$, and second half $S = A^0[(n/2+1)..n]$

Further define $Q_1 = F[1..n/4], Q_2 = F[(n/4+1)..n/2]$ be the first and second quarter of A^0. $Q_3 = S[1..n/4], Q_4 = S[(n/4+1)..n/2]$ are the third and last quarters. An immediate conclusion can be drawn is that Q_1, Q_2, Q_3, Q_4 are all sorted arrays because sub-arrays of

sorted array are still sorted.

Since $n > 2$ on line 1 is true, only line $2 - 6$ will be executed.
First consider line 2-3. Notice that when $1 \geq i \geq \frac{n}{4}$, by our definition of Q_2, Q_3, we have $A[i + n/4] = Q_2[i]$ and $A[i + n/2] = Q_3[i]$. Thus, executing line 3 is the effect of swapping Q_2 and Q_3. If $A = Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4$ where "$\cdot$" denote array concatenation, after line 3, $A^3 = Q_1 \cdot Q_3 \cdot Q_2 \cdot Q_4$.

Now consider line 4, which performs AUX on $A^3[1..n/2] = Q_1 \cdot Q_3$. Since Q_1, Q_3 are equal lengths of some power of 2, sorted (justified above), by our inductive hypothesis, after the execution, the first half of A^4, or $B = A^4[1..n/2]$ will be sorted version of $Q_1 \cdot Q_3$.
Now consider line 5, which performs AUX on $A^4[(n/2 + 1)..n] = Q_2 \cdot Q_4$ (line 3 does not affect the second half of A^3). By the same reasoning, after the execution, the second half of A^5, or $C = A^5[(n/2 + 1)..n]$, will be sorted version of $Q_2 \cdot Q_4$.

Since line 5 does not affect the first half as well, $A^5 = B \cdot C$ where B, C are defined above. For clarity, further define $F_B = B[1...n/4], F_C = C[1...n/4]$ be first half of B, C, respectively. $S_B = B[(n/4 + 1)..n/2], F_C = C[(n/4 + 1)..n/2]$ be second half of B, C. We can rewrite A^5 as $A^5 = F_B \cdot S_B \cdot F_C \cdot S_C = \text{sorted}(Q_1 \cdot Q_3) \cdot \text{sorted}(Q_2 \cdot Q_4)$.

We will prove 2 crucial facts about A^5:
Lemma 1. for every $x \in F_B, y \in (S_B \cdot F_C), x \leq y$.
Proof.
Let $x \in F_B, y \in (S_B \cdot F_C)$ be arbitrary.
By our definition of F_B, $x \in B[1..n/4]$. By def of B, $x \in Q_1$ or $x \in Q_3$. Assume with out the loss of generality that $x \in Q_1$. We can do this because Q_1, Q_3 corresponds to first half of F, S, where F, S by themselves are completely arbitrary sorted arrays.
Since $y \in (S_B \cdot F_C)$, by our definition, its possible for y to be in any of Q_1, Q_2, Q_3, Q_4.
If $y \in S_B, y \in (Q_1 \cdot Q_3)$. If $y \in F_C, y \in (Q_2 \cdot Q_4)$
Case 1: $y \in S_B$. Since we know that B is the sorted version of $Q_1 \cdot Q_3$, and S_B comes after F_B ($B = F_B \cdot S_B$). Hence, it is trivial that $x \leq y$ as y must has a greater index in B compare to x and B is in non-decreasing order.

Case 2: $y \in F_C$ and $y \in Q_2$. Since Q_1, Q_2 are from the same sorted list $A^0[1..n/2]$ where Q_2 comes after Q_1, hence $x \leq y$.

Case 3: $y \in F_C$ and $y \in Q_4$. We will assume for contradiction that $x > y$. By our the definition of Q_4, we know that $y = S[i]$ for some $i \geq n/4 + 1$ (S is the second half of A^0 and is sorted). Also for all $z \in S[1..n/4]$, $z \leq y$ as S is sorted.
Concatenate the two inequality, for all $z \in S[1..n/4]$, $z \leq y < x$. Since $S[1..n/4] = Q_3$, and B contains all elements from Q_1 and Q_3, sorted in non-decreasing order, this suggests that every elements in Q_3 will be placed before x. However, since F_B is the same size of Q_3, there is a contradiction with $x \in F_B$.
Hence, it must be the case that $x \leq y$.
We have exhausted all cases, thus $x \leq y$ is true.

Since $x \in F_B, y \in (S_B \cdot F_C)$ are arbitrary $\forall x \in F_B . \forall y \in (S_B \cdot F_C), x \leq y$ by generalization.

(A little more special notes about proving when $x \in Q_3$: the reason why we can assume without the loss of generality is that if $x \in Q_3$, by can prove by symmetry using the prove assuming $x \in Q_1$. we can change every Q_2 in case 2 to Q_4 and the argument holds. Similarly, change every Q_4 to Q_2 in case 3 and we are done.
Essentially, case 1 says if y is after x and is in the after half of A^5, line 4 ensures $y \leq x$ because each half is sorted. Case 2 says if y is in the second half of A^5 but is in the the same half at the beginning in A^0, by the precondition $y \leq x$. Case 3 says when swapping Q_3 and Q_2 and sort the first half, the new first quarter will stay at most as large as Q_3. Hence since Q_4 is greater than Q_3 before, it must be greater than F_B after as well.

Above paragraph in parenthesis is not rigorous and serves only as additional explanation.)

Lemma 2. for every $x \in S_C, y \in (S_B \cdot F_C), y \leq x$.
Proof.
Let $x \in S_C, y \in (S_B \cdot F_C)$ be arbitrary.
Since S_C is the second half of C, and C is the sorted version of $Q_2 \cdot Q_4$, its either the case that $x \in Q_2$ or $x \in Q_4$. For the same reasoning given above, assume without the loss of generality that $x \in Q_2$ (where we can appeal to symmetry if $x \in Q_4$).
Case 1: $y \in F_C$.
Since $x \in S_C$ and $C = F_C \cdot S_C$ is a sorted list, it follows that $y \leq x$ as elements in the second half (S_C) have higher indices than those of first half F_C.
Case 2: $y \in S_B$ and $y \in Q_1$.
Recall that Q_1, Q_2 are the first and second half of the sorted list, F. With the above reasoning, $y \in Q_1$ must be smaller than or equal to $x \in Q_2$.
Case 3: $y \in S_B$ and $y \in Q_3$.
Assume for contradiction that $y > x$. Since $y \in Q_3$ is smaller than or equal to all $z \in Q_4$ (where there are $n/4$ of them), by our definition of $C = \text{sorted}(Q_2 \cdot Q_4)$, there are additional $n/4$ elements other than y that is greater than x and are in C. However, $x \in S_C$ says x is in the last $n/4$ elements of C. In other words, there can be at most $(n/4) - 1$ elements other than y that is greater than x in C.
Thus a contradiction is reached. This means $y \leq x$ must be true.
We have exhausted all cases, thus $x \leq y$ is true.
Since $x \in S_C, y \in (S_B \cdot F_C)$ are arbitrary $\forall x \in S_C . \forall y \in (S_B \cdot F_C), y \leq x$ by generalization.

(Special Note: Lemma 1 and 2 have very similar idea. I used two different languages when proving each of them in case one of them is confusing. However, these two explanation essentially says the same thing.)

Now consider line 6, which calls AUX on the middle two quarters of $A^5 = F_B \cdot S_B \cdot F_C \cdot S_C$. Since B, C are sorted lists, their sub-lists S_B, F_C are all sorted too (with length a power of 2). Thus, $S_B \cdot F_C$ satisfies the precondition. Hence $\text{AUX}(A[(\frac{n}{4}+1)..\frac{3n}{4}], n/2)$ will sort the middle two quarters $S_B \cdot F_C$. We will denote $M = \text{sorted}(S_B \cdot F_C)$. After the execution, $A^6 = F_B \cdot M \cdot S_C$ where $|F_B| = |S_C| = n/4$ and $|M| = n/2$.

Since during line $4, 5, 6$, the only mutation to list is merging with AUX call. By our inductive hypothesis, the multiset of A^i is preserved in those lines. Hence after line 6 is executed, the multiset is unchanged.
Also, our inductive hypothesis guaranteed that line $4, 5, 6$ will finish in finite steps. Hence the program halts after executing line 6.

We will finally show that A is sorted in non-decreasing order.
Claim. if $x = A^6[i], y = A^6[j]$ are some elements of A^6 and $i < j$, then $x \leq y$.
Proof.
Let $x = A^6[i], y = A^6[j] \in A^6$ be arbitrary and assume $i < j$.
Recall that $A^6 = F_B \cdot M \cdot S_C$ where $M = \text{sorted}(S_B \cdot F_C)$.
Case 1: $(x, y \in F_B)$ OR $(x, y \in M)$ OR $(x, y \in S_C)$.
Since F_B, M, S_C are each sorted lists (in non-decreasing order). $i < j$ implies $x \leq y$.
Case 2: $x \in F_B$ and $y \in M$.
By definition that $M = \text{sorted}(S_B \cdot F_C)$, it follows that $y \in (S_B \cdot F_C)$. By lemma 1 and specialization, we have $x \leq y$.
Case 3: $x \in M, y \in S_C$.
Since $x \in (S_B \cdot F_C), y \in S_C$, by lemma 2 and specialization, we have $x \leq y$.
Case 4: $x \in F_B$ and $y \in S_C$.
By our construction of M, it is non-empty and there is some $z \in M$ where $z \in (S_B \cdot F_C)$.
By lemma 1, $x \leq z$. By lemma 2, $z \leq y$. Concatenation gives us $x \leq y$.
We have exhausted all cases where $i < j$. Hence $x \leq y$ is true.
Since $x = A^6[i], y = A^6[j] \in A^6$ are arbitrary, we proved the desired claim.

By direct line-by-line code analysis of the algorithm, we can conclude that if AUX(A[1..n], n) is performed, then it eventually halts, at which time A is sorted in non-decreasing order and the multiset of elements in A is unchanged.

Since $A[1..n]$ is an arbitrary list of length n satisfying the precondition of AUX, $P(n)$ is true.
By the principle of structural induction and $n \in \mathcal{P}_2 \setminus \{1\}$ is arbitrary: $\forall n \in \mathcal{P}_2 \setminus \{1\}.P(n)$.
As the precondition says n is a power of 2 greater than 1, we proved that AUX is correct.

4. Prove that SRT is correct. You may assume that AUX satisfies the specifications in your answer to question 2.

Solution.

First recall the preconditions and postconditions for such merge algorithm:
Preconditions:
$A[1..n]$ is an array with elements from a totally ordered set.
$n = 2^i$ for some integer $i \geq 1$.
Postconditions:
If **SRT**$(A[1..n], n)$ is performed, when it halts, A is sorted in non-decreasing order and the multiset of elements in A is unchanged.

For $n \in \mathcal{P}_2$. Let $Q(n) =$"for all arrays $A[1..n]$ satisfying the preconditions, if SRT(A[1..n], n) is performed, then it eventually halts, at which time A is sorted in non-decreasing order and

the multiset of elements in A is unchanged."

We will prove by structural induction that $\forall n \in \mathcal{P}_2.Q(n)$.

Proof.
Let $n \in \mathcal{P}_2$ be arbitrary.
Let $A[1..n]$ be an arbitrary array of elements from a totally ordered set with length n.
Base case: assume $n = 1$.
The test on line 1 fails, algorithm terminates immediately.
So A is unchanged, trivially A is sorted.
Inductive Case: assume $n > 1$ and assume $Q(n/2)$ is true.
The test on line 1 is true so program proceeds to executing line 2-4.
Consider line 2:
Since $Q(n/2)$ is true, SRT$(A[1..\frac{n}{2}], n/2)$ will correctly sort $A[1..\frac{n}{2}]$ into non-decreasing order.
This line will finish in finite step and multiset is unchanged.
Consider line 3:
SRT$(A[(\frac{n}{2}+1)..n], n/2)$ will correctly sort $A[1..\frac{n}{2}]$ into non-decreasing order.
This line will finish in finite step and multiset is unchanged.
Consider line 4:
As $n > 1$, and is a power of 2, $n \in \mathcal{P}_2 \setminus \{1\}$ is met .
First half $(A[1..\frac{n}{2}])$ and second half $(A[(\frac{n}{2}+1)..n])$ of A are sorted in non-decreasing order.
Hence the preconditions of AUX is met. We proved that AUX is a correct algorithm, thus line 4 will merge A to a sorted list in non-decreasing order and halts.
Since $A[1..n]$ are arbitrary, $Q(n)$ is true.
Since n are arbitrary, by the principle of structural induction, we proved $\forall n \in \mathcal{P}_2.Q(n)$.
Therefore, SRT is correct.