CSC236, A3

Youngsin Ryoo

Q1

(a) Let n = len($A[0 : e + 1]$)
Let a = constant running time for lines 13, 14
Let b = constant running time for lines 4, 5, 6, 11
Let c = constant running time for lines 8, 9
$T(n-1)$ comes from line 3.

$$T(n) = \begin{cases} a, & n = 0 \\ T(n-1) + c(n-1) + b, & n > 0 \end{cases}$$

(b) **Step 1.**
B is sorted and B contains all elements in $A[0 : e]$ and nothing else.
Step 2.
$LI(k)$: If the loop iterates at least k times then
(a) C_k is sorted (non-decreasing).
(b) C_k contains all elements that are in $B[0 : i_k]$ and nothing else.
(c) All elements in $B[i_k : len(B)]$ are greater than or equal to all elements in C_k.
(d) $i_k \leq$ len(B)
Step 3.
Assuming the loop precondition, prove for all $k \in \mathbb{N}$, $LI(k)$ holds, using induction.
Base Case.
Let $k = 0$.
We know that $C = []$. Since $i = 0$ and there's no element in $B[0 : i_0]$, C_0 is sorted and contains all elements in $B[0 : i_0]$.
Also, $B[i_0 : len(B)]$ are greater than or equal to all elements in C_0 because $C = []$.
Finally, because $i_0 = 0$, $i_0 \leq len(B)$.
So, $LI(k)$ holds.
Inductive Step.
Let $k \in \mathbb{N}$. Assume $LI(k)$ holds. [IH]
WTP: $LI(k + 1)$ holds.
Assume $k + 1$ iterations exist.
By line 9, $i_{k+1} = i_k + 1$
Since $k + 1$ iterations exist, the loop condition must hold after k iterations, and so $i_k < len(B)$.

So, $i_{k+1} = i_k + 1 \leq len(B)$
Thus, $\leq len(B)$.
Since $k+1$ iterations exist, the loop conditions hold after k iterations, which means that C_k is sorted, contains all elements that are in $B[0 : i_k]$ and nothing else, and all elements in $B[i_k : len(B)]$ are greater than or equal to all elements in C_k.
At the start of $k+1$ iterations, C_{k+1} would be sorted because C would append $B[i_{k+1}]$ when $i_{k+1} < len(B)$ and $B_{k+1} < 1$, which both would be true by the assumption that $k+1$ iterations exist.
Then, C_{k+1} will contain all elements that are in $B[0 : i_{k+1}]$ and nothing else.
Finally, all elements in $B[i_{k+1} : len(B)]$ would be greater than or equal to all elements in C_{k+1} because C_{k+1} is sorted and $B[i_{k+1} : len(B)]$ is sorted as $B[i_k : len(B)]$ is sorted.
Step 4.
The loop on line 7 terminates.
Step 5.
$P(n)$: For all A, if A is a non-empty list of numbers and e is a natural number and $e < len(A)$ and $n = len(A[0 : e+1])$, then $Sort(A, e)$ terminates and returns a list that contains all the elements in $A[0 : e+1]$ sorted in non decreasing order.
WTP: For all $n \in \mathbb{N}$, if $n \geq 1$, then $P(n)$ holds.
Base Case.
Let $n = 1$.
Since $n = 1$, e should be 0, so the if-condition on line 2 fails and $Sort(A, e)$ runs line 14.
Thus, it terminates and returns $A[0 : e+1]$, which is sorted because it only contains one element.
Inductive Step.
Let $n \in \mathbb{N}$, $n > 1$.
Assume for all $j \in \mathbb{N}$, $1 \leq j < n$, $P(j)$ holds. [IH]
WTP: $P(n)$ holds.
Since $n > 0$, the if-condition on line 2 is satisfied and $Sort(A, e)$ runs lines 3-12.
Note that $1 \leq len(A[0 : e]) < n$.
Then, by IH, $Sort(A, e-1)$ terminates and assign to B a sorted version of $A[0 : e]$.
So the precondition of the while loop holds and we can say that the loop terminates after t iterations, and after the termination of the loop, $LI(t)$ holds.
Case 1. Assume $\neg(i_t < len(B))$ holds.
Then $B[i_t : len(B)]$ is empty.
By $LI(t)$, C_t is sorted and contains all elements in $B[0 : i_t] = B[0 : len(B)] = B$ and nothing else. ... ①
By $LI(t)$, all elements in $B[i_t : len(B)]$ are greater than or equal to all elements in C_t.
At line 11, C_t appends $l = A[e]$ at the end and C_t is still sorted because before the loop terminates $B[i] < l$.
Since C_t is sorted and $B[i_t : len(B)]$ is empty, $C_t + B[i_t : len(B)]$ is sorted. ... ②
With ① and ②, the post-condition holds.
Also, the program terminates at line 12.
Thus, $P(n)$ holds.
Case 2. Assume $\neg(B[i_t] < l)$ holds.

Then $B[i_t] \geq l$. i.e. $B[i_t] \geq A[e]$.
By $LI(t)$, C_t is sorted and contains all elements in $B[0 : i_t]$ and nothing else.
Also, by $LI(t)$, all elements in $B[i_t : len(B)]$ are greater than or equal to all elements in C_t.
If $B[i_t] \geq A[e]$, the loop terminates at line 9 and C_t will append $A[e]$ that is less than or equal to $B[i_t]$, instead of $B[i_t]$, so that the elements are sorted in non decreasing order.
Since C_t is sorted and $B[i_t : len(B)]$ are greater than or equal to all elements in C_t, $C_t + B[i_t : len(B)]$ is also sorted.
Therefore, the post condition holds and $P(n)$ holds. ■