Administrativia: no discussions, no extra material consulted

1 Problem

For any $\Delta \in \mathbb{Z}^+$, let $R(A, \Delta(first, last)) =$"RSUM$(A, first, last)$ is the largest sum of consecutive elements in the sequence of integers stored in the array $A[first..last]$ for any nonnegative integers $first$ and $last$ such that $last - first + 1 = \Delta$, $first \leq last$, and $last \leq len(A)$".

Preconditions:

1. $first \in \mathbb{N}$, $last \in \mathbb{N}$
2. $first \leq last$
3. $last \leq len(A)$
4. for any $i \in [first, last] \cap \mathbb{N}$, $A[i] \in \mathbb{Z}$

Postconditions:

1. $A, first, last$ are unchanged
2. RSUM$(A, first, last)$ is the largest sum of consecutive elements in the sequence of integers stored in the array $A[first..last]$

Let $f \in \mathbb{N}$, $l \in \mathbb{N}$ be such that $f \leq l$ and $l \leq len(A)$.

We proceed by induction on $\Delta = l - f + 1$, which corresponds to the length of the sequence of integers stored in the array $A[f..l]$.

Since $l \geq 0$, $f \geq 0$ by construction, then $\Delta = l - f + 1 \geq 1$.

Suppose first that $\Delta = 2^0 = 1$, and thus there is only one element if the sequence corresponding to $A[f..l]$.

Therefore, $l = f$, and hence the condition on line 2 is satisfied.

Note that an array $A[f..l]$ consists of one integer $A[f] = A[l]$. Therefore, if this integer is negative, the largest sum of consecutive elements in $A[f..l]$ is the empty sum, which is equal to zero. If the integer is nonnegative, then the largest sum of consecutive elements is equal to the value of the only element.

On line 1, $best$ was assigned to zero, and hence the if-statement on line 3 and the conditional assignment on line 4 guarantee, corresponding to our reasoning in the prevous paragraph, that the largest sum of consecutive elemens is returned on line 24. Therefore, $R(A, \Delta(f, l))$ holds for $\Delta(f, l) = 1$ by generalisation.

Suppose now that $f \neq l$.

Assume $R(A, \Delta(f, l))$ holds for all $\Delta = i$, where $i \in [1, 2^k] \cap \mathbb{Z}^+$ and $k \in \mathbb{N}$.

Consider the case when $\Delta = 2^k + 1$.

Consider an arbitrary $l \in \mathbb{N}, f \in \mathbb{N}$ such that $f \leq l$ and $l \leq len(A)$, and $l - f + 1 = 2^k + 1$. Therefore, $l = f + 2^k$.

In the algorithm for $A[f..l]$, the else-statement is executed on line 5, because $l \neq f$. Therefore, after the execution of the line 5, we have $best \leftarrow 0$ and, since $\left\lfloor \frac{l+f}{2} \right\rfloor = \left\lfloor \frac{f+f+2^k}{2} \right\rfloor = \left\lfloor 2^{k-1} + f \right\rfloor = 2^{k-1} + f$, then $mid \leftarrow 2^{k-1} + f$.

After line 7, $b \leftarrow 0$ and $i \leftarrow 2^{k-1} + f + 1$.

Note that steps on line 9, 10 and 11 are executed only for $i \in [2^{k-1} + f + 1, 2^k + f] \cap \mathbb{N}$, because $l = f + 2^k$ and $i \leq l$, while $i \geq mid + 1$ (as guaranteed by line 11), if and only if $i \in [2^{k-1} + f + 1, 2^k + f] \cap \mathbb{N}$.

Therefore, there are at most $2^k + f - 2^{k-1} - f - 1 + 1 = 2^{k-1}$ executions of each line inside the loop on lines 9-11, and thus the loop initiated on line 8 is not infinite.

Lines 9 to 11 ensure that after the execution of the loop *best* is equal to the largest sum of consecutive elements in A from $mid + 1$ to $last$, so that all the sums checked include the contribution of $A[mid + 1]$, for the following reason. Before the first iteration, b is equal to 0. After the first iteration is executed, in each new iteration after the update on line 9 b is equal to the sum of the elements in the array with the index less than or equal to the current value of i, starting with $i = mid + 1$, as guaranteed by line 7. Line 10 then ensures that *best*, which was equal to 0 before the first iteration, is equal to the greatest running total of the elements with the index less than or equal to the current value of i. The value of i is then updated, and the procedure repeats until all the sums of consecutive elements starting with $A[mid + 1]$ are checked.

On line 12, 0 is assigned to $best'$, and on line 13 b is reset to 0, while on line 14 i gets the value of mid.

Note that steps on line 16, 17 and 18 are executed only for $i \in [f, 2^{k-1} + f] \cap \mathbb{N}$, because $mid = 2^{k-1} + f$ and $i \geq f$, while $i \leq mid$ (as guaranteed by line 18), if and only if $i \in [f, 2^{k-1} + f] \cap \mathbb{N}$.

Therefore, there are at most $2^{k-1} + f - f + 1 = 2^{k-1} + 1$ executions of each line inside the loop on lines 16-18, and thus the loop initiated on line 15 is not infinite.

We prove now that the loop initiated on line 15 and defined on lines from 16 to 18 returns the maximum sum of consecutive elements chosen from the elements of the array from $A[1]$ to $A[mid]$ so that all the sums checked include the contribution of $A[mid]$.

After the first iteration of the loop is executed, in each new iteration after the update on line 16 b is equal to the sum of elements in the array with the index greater than or equal to the current value of i. Line 17 then ensures that $best'$, which was equal to 0 before the first iteration, is equal to the greatest running total of the elements with the index greater than or equal to the current value of i up to the index equal to mid. In this way, all the sums which are assigned to b after the first iteration include the contribution of $A[mid]$. The value of i is then updated, and the procedure repeats until all the possible sums of consecutive elements having the highest possible index of mid and the least possible index of 1, necessarily including the contribution of $A[mid]$, are checked.

On line 19, the best value is updated so that it is equal to the maximum sum of consecutive elements in $A[f..l]$ such that the sum contains the contributions of $A[mid]$ and $A[mid+1]$, since before the execution *best* is equal to the maximum sum of consecutive elements from $A[mid + 1]$ to $A[l]$ and $best'$ is equal to the maximum sum of consecutive elements from $A[1]$ to $A[mid]$.

By inductive hypothesis, $R(A, \Delta(f, mid))$ and $R(A, \Delta(mid+1, l))$ hold, since $mid-f+1 = 2^{k-1} + 1 \in [1, 2^k] \cap \mathbb{Z}^+$ and $l - mid - 1 + 1 = f + 2^k - 2^{k-1} - f - 1 + 1 = 2^{k-1} \in [1, 2^k]$. Therefore, line 20 assigns to b the maximum sum of consecutive elements in $A[f..mid]$, and line 23 assigns to b the maximum number of consecutive elements in $A[mid + 1..l]$. Thus, line 21 checks if the maximum sum of consecutive elements in $A[f..l]$ containing the contributions of $A[mid]$ and $A[mid + 1]$ is less than RSUM(A, f, mid). After the execution of line 21, there is only one case left to check, namely whether the maximum sum of consecutive elements not containing elements with indices from 1 to mid is greater than the sum containing their contributions, the check for which is performed by line 23. In this way, all the three cases are verified, which are such that:

- the maximum sum of consecutive elements in $A[f..l]$ contains the contribution of the elements $A[mid]$ and $A[mid+1]$
- the maximum sum of consecutive elements in $A[f..l]$ contains the contribution of $A[mid]$, but not of $A[mid+1]$ (and thus this sum is equal to RSUM(A, f, mid))
- the maximum sum of consecutive elements in $A[f..l]$ contains the contribution of $A[mid+1]$, but not of $A[mid]$ (and thus this sum is equal to RSUM$(A, mid+1, l)$)

Hence, line 24 returns the maximum sum of consecutive elements in $A[f..l]$.

Moreover, by inductive hypothesis the execution of lines 20 and 22 is finite.

Note that $A, first, last$ were not changed after the execution of line 24.

Therefore, $R(A, \Delta(f, l))$ holds by generalisation.

Thus, for any $first \in \mathbb{N}$, $last \in \mathbb{N}$ such that $first \leq last$ and $last \leq len(A)$ we have shown that $R(A, \Delta(first, last))$ by induction.

Since the preconditions and postconditions are satisfied, while the algorithm terminates, we have shown that RSUM is totally complete.