Correctness Proof for the $O(n^2)$ Algorithm for 3SUM

Douglas R. Stinson
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

January 9, 2019

In this note, I will give a proof of correctness for the Qudaratic3SUM algorithm presented on slide # 23. For simplicity, we are assuming that the elements in $S$ are distinct.

It is clear that any solution found by the algorithm is valid, so we just need to show that the algorithm will find all the solutions. Suppose that $A[i_0] + A[j_0] + A[k_0] = 0$, where $i_0 < j_0 < k_0$. The algorithm should find this solution when $i = i_0$ in the for loop. We assume that the algorithm does not find this solution, and then derive a contradiction.

When the iteration $i = i_0$ of the algorithm terminates, we have $j = k = \ell$, say. We consider three cases.

case 1. Suppose $\ell \leq j_0$. At some point in time before the termination of this iteration, $k$ must have been decremented from $k_0$ to $k_0 - 1$ since $\ell \leq j_0 < k_0$. At that point in time, $i = i_0$, $j \leq \ell \leq j_0$ and $k = k_0$. Then


Since $k$ is decremented only when $A[i] + A[j] + A[k] \geq 0$, it must be the case that $A[i] + A[j] + A[k] = 0$. This happens if and only if $(i, j, k) = (i_0, j_0, k_0)$. In this case, the solution $(i_0, j_0, k_0)$ would have been reported by the algorithm, which is a contradiction.

case 2. Suppose $\ell \geq k_0$. This case is handled by an argument similar to case 1.

case 3. Suppose $j_0 < \ell < k_0$. At some point in time before the termination of this iteration, $k$ must have been decremented from $k_0$ to $k_0 - 1$ since $\ell < k_0$. In order for $k$ to be decremented, $A[i] + A[j] + A[k] \geq 0$ at this point in time. We have $i = i_0$ and $k = k_0$, so $A[i] + A[j] + A[k] \geq 0$ only if $j \geq j_0$. If $j = j_0$, then $A[i] + A[j] + A[k] = 0$ and the solution $(i_0, j_0, k_0)$ would have been reported by the algorithm. If $j > j_0$, then there was a time earlier in this iteration when $j$ was incremented from $j_0$ to $j_0 + 1$. At this time, $k \geq k_0$, so


However, $j$ is not incremented unless the sum is $\leq 0$. Hence we conclude that $(i, j, k) = (i_0, j_0, k_0)$ and the solution $(i_0, j_0, k_0)$ would have been reported by the algorithm.