

Proving the Partial Correctness of a Simple Algorithm with a While Loop

A Suggested Exercise

About This Exercise

This exercise is intended to help you to practice and improve your skills in proving that a simple algorithm with a `while` loop is partially correct.

Problems To Be Discussed in This Tutorial

Consider the following computational problem.

Sum of Array Elements

Precondition: An integer array A with some positive length n is given as input.

Postcondition: The value

$$\sum_{i=0}^{n-1} A[i]$$

is returned as output.

Consider, as well, the following algorithm (as an algorithm that can be used to solve the above problem).

```
integer arraySum ( integer[] A ) {  
  1. integer sum := A[0]  
  2. integer i := 0  
  3. while (i < A.length - 1) {  
  4.   i := i + 1  
  5.   sum := sum + A[i]  
  }  
  6. return sum  
}
```

1. Prove that the following is a **loop invariant** for the `while` loop in this algorithm.

Loop Invariant:

- (a) `A` is an input integer array with some positive length n .
- (b) `i` is an integer variable such that $0 \leq i \leq n - 1$.
- (c) `sum` is an integer variable such that

$$\text{sum} = \sum_{j=0}^i A[j].$$

2. Use this, as needed, to prove that the `arraySum` is partially correct (when considered as an algorithm for the “Sum of Array Elements” problem).