CSC236 Tutorial 8

Sample Solutions

1. Give a proof of correctness for the program below with respect to its given specification.
   Hint: For any integers a, b such that $a + 1 < b$,

$$a < \lfloor \frac{a+b}{2} \rfloor < b.$$

Precondition: A is a list of integers, $0 \le p < q \le len(A)$.
Postcondition: Return the index of a minimum element in $A[p : q]$. That is, return a number i such that $p \le i < q$ and $A[i]$ is the smallest integer of $A[p : q]$.

```
    def IndexMin(A, p, q):
1.      if p + 1 == q:
2.          return p
3.      else:
4.          m = ⌊(p+q)/2⌋
5.          j = IndexMin(A, p, m)
6.          k = IndexMin(A, m, q)
7.          if A[j] ≤ A[k]:
8.              return j
9.          else:
10.             return k
```

Solution: For $n \in \mathbb{N}$, we define the predicate $P(n)$ as follows.
$P(n)$: if A is a list of integers, $0 \le p < q \le len(A)$, and $n = q - p$, then $IndexMin(A, p, q)$ terminates and returns the index of a minimum element in $A[p : q]$.

Note: We define $q - p = len(A[p : q])$ as our input size.

By complete induction, we prove $P(n)$ holds for all integers $n > 0$.
Then correctness follows.

Base case: Let $n = 1$. That is, $A[p : q]$ contains just 1 element. Then p is the index of the only, and hence smallest, element in $A[p : q]$. By lines 1-2, $IndexMin(A, p, q)$ returns p as wanted.

Induction Step: Let $n > 1$, i.e., $A[p : q]$ contains more than one elements. Suppose $P(j)$ holds whenever $1 \le j < n$. **[IH]**
WTP: $P(n)$ holds.

For $n = q - p > 1$, the condition on Line 1 is false.
So $IndexMin(A, p, q)$ runs Lines 4-10.
By line 4, $m = \lfloor \frac{p+q}{2} \rfloor$.
By Hint, $p < m < q$.
So $0 \le p < m \le len(A)$ and $1 \le m - p < n$. (1)
Also $0 \le m < q \le len(A)$ and $1 \le q - m < n$. (2)
By (1), IH and Line 5 and Line 6, j is the index of a smallest element in $A[p : m]$, and k is the index

of a smallest element in $A[m:q]$.
Thus j is the index of a smallest element in $A[p:q]$ if $A[j] \leq A[k]$, and k is the index of a smallest element in $A[p:q]$ if $A[j] > A[k]$.
Therefore by Lines 8 and 10, the index of a smallest element in $A[p:q]$ is returned as wanted.

2. Consider the following program.
Precondition: A is nonempty list of integers.
Postcondition: The elements of A are rearranged in sorted (nondecreasing) order.

```
def Sort(A):
    k = 0
    while k < len(A) − 1:
        j = IndexMin(A, k, len(A)) # see question 1 for specification of IndexMin
        A[k], A[j] = A[j], A[k] # swap A[k] and A[j]
        k = k + 1
```

(a) Give an appropriate loop invariant for the purpose of proving both partial correctness and termination for the above program with respect to its given specification. For this part a proof is not required.

(b) Define an appropriate loop measure for the purpose of proving termination. For this part a proof is not required.

(c) Assume your loop invariant from part (a) is correct and use it to prove partial correctness.

Solution:

(a) Let $LI(j)$ denotes the assertion that if the loop is executed at least j times, then

i. $0 \leq k_j \leq len(A) - 1$.

ii. The elements of A are rearranged so that $A[0 : k_j + 1]$ is sorted in nondecreasing order.

iii. The elements of A are rearranged so that every elements of $A[0 : k_j]$ is less than or equal to every element of $A[k_j : len(A)]$.

(b) $m_j = len(A) - k_j$. [$len(A) - 1 - k_j$ also works.]

(c) Suppose the loop terminates, and consider the values of A, k on exit.
By LI(i), $k_j \leq len(A) - 1$.
By the exit condition, $k_j \geq len(A) - 1$.
Hence,

$$k_j = len(A) - 1. \qquad (\star)$$

By LI(ii), the elements of A are rearranged so that $A[0 : k_j + 1]$ are sorted. By $(\star)$, $A[0 : k_j + 1]$ is just all of A. Thus the elements of A are rearranged so the A is sorted as wanted.