CSCI 430: Homework 2

Annabelle Cormia

September 7, 2017

Tyler Archer and I collaborated on this assignment.

1 Chapter 2.1 (2.1-1 through 2.1-4)

1. See Figure 1.

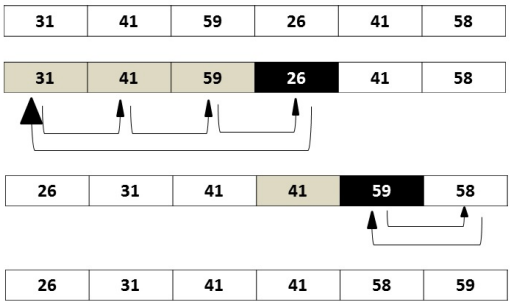

Figure 1: Illustration of the operation of INSERTION-SORT on the array A = (31, 41, 59, 26, 41, 58).

2. INSERTION-SORT (A)

```
1       for j = 2 to A.length
2           key = A[j]
3           // Insert A[j] into the sorted sequence A[1.. j - 1].
4           i = j - 1
5           while i > 0 and A[i] > key
6               A[i + 1] = A[i]
7               i = i - 1
8           A[i + 1] = key
```

3. LINEAR-SEARCH (A)

```
1       for i = 1 to A.length
```

```
2           while v ≠ A[i]
3                 i = i + 1
4           if v = A[i]
5                 return i
6     if i > A.length
7           return NIL
```

Initialization: We start by showing that the loop invariant holds before the first loop iteration, when i = 1. The loop invariant is vacuously true because it is empty.

Maintenance: Next, we must show that each iteratino maintains the loop invariant. The while loop works by moving through A until it finds a value whose index equals v. Each time it does not find a value whose index matches v, it increments i to keep checking. If v equals A[i], then i is returned. If i exceeds the length of A, then NIL is returned.

Termination: Finally, we examine what happens when the loop terminates. The condition causing the for loop to terminate is that i > A.length which results in NIL. However, if a value i is found that equals v, that value is returned. Therefore, we conclude that the output is either an index i such that v = A[i] or the value NIL if v does not appear in A. Thus the algorithm is correct.

4. Adding two binary integers each of length n will result in an (n+1)-element array C. Suppose the two integers are stacked on top of each other so that the right-most element of each array is lined up. The carry digit will begin at 0. A, B, and carry are added together to produce sum for each column of digits. If sum is 0, then the carry element for the next column to be summed is 0 and the digit going into the array C is also 0. Array C holds the final sum of arrays A and B, and it has length n+1 to account for a surplus carry at the end of the addition. If the sum is 1, then the carry element is 0 and the next digit in the C array is 1. If the sum is 2, then the carry is 0 and the next digit in C is 1. If the sum is 3, then both the carry and next element in C are 1. Variable i is then decremented until the end of the array when the final carry is placed into the remaining spot in the C array.

```
ADDING-BINARY-INTEGERS (A + B)
1     carry = 0
2     for n to i=1 // n is length of both A and B
3           sum = A[i] + B[i] + carry
4           if sum = 0
5                 carry = 0
6                 C[i + 1] = 0
7           if sum = 1
```

```
8                   carry = 0
9                   C[i +1]= 1
10              if sum = 2
11                  carry = 0
12                  C[i + 1] = 1
13              if sum = 3
14                  carry = 1
15                  C[i + 1] = 1
16              i−−
17          C[1] = carry
```