Cormen 2.1 exercises series.

Iván Alejandro Cruz Tole.

10.09.2018

1 First exercise.

Using Figure 2.2 as a model, illustrate the operation of INSERTION SORT on the sequence A = 31; 41; 59; 26; 41; 58.

1. 31; 41; 59; 26; 41; 58 - sort A[0:2].
2. 31; 41; 59; 26; 41; 58 - sort A[0:3].
3. 26; 31; 41; 59; 41; 58 - sort A[0:4].
4. 26; 31; 41; 41; 59; 58 - sort A[0:5].
5. 26; 31; 41; 41; 58; 59 - sort A[0:6].

2 Second exercise.

Rewrite the INSERTION SORT procedure to sort into nonincreasing instead of nondecreasing order. Given a sequence 'A'...

```
for (i = 1; i < A.length; i++):
  key = A[i]
  for (j = i - 1; j > -1; j–):
    if (A[j] < key):
      A[j+1] = A[j]
    else:
      A[j+1] = key
      break
```

3 Third exercise.

Write pseudocode for linear search, which scans through a sequence A, looking for a value v. Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

```
i = 0;
while (i < A.length and A[i] != v):
  i++
if (A[i] == v):
  return i
else:
  return Nil
```

The correctness of the algorithm is clear given that for any i between 0 and A.length we'll always have that all the previous indexes of the sequence have been already covered, meaning that v isn't in any of them, but possibly in i. This possibility is covered by the if statement below the while statement.

4 Fourth exercise.

Consider the problem of adding two n-bit binary integers, stored in two n-element arrays 'A' and 'B'. The sum of the two integers should be stored in binary form in an (n + 1)-element array 'C'. State the problem formally and write pseudocode for adding the two integers.

Input: Two arrays (A and B), each of n elements and each element a 1 or a 0.

Output: An array (C) of n+1 elements representing the binary sum of the input arrays.

```
carry = 0
for (i = n - 1; i > -1; i– ):
  if (A[i] + B[i] + carry > 1):
    carry = 1
    C[i+1] = 0
else:
    carry = 0
    C[i+1] = 1
C[0] = carry
```