Benha University

Computer
ECE 001

Computer Systems Engineering
Electrical Engineering Department

Faculty of Engineering
(at Shoubra)

Sheet 2

I Solve the following *Review Problems* from *Computer Science: An Overview*:

- **5.22**
  The following algorithm is designed to print the beginning of what is known as the *Fibonacci* sequence. Where is the body of the loop? The initialization step for the loop control? The modification step? The test step? What list of numbers is produced?

```
F0 ← 0;
F1 ← 1;
while (F1 < 100) do {
    print F1;
    Temp ← F0;
    F0 ← F1;
    F1 ← F0 + Temp;
}
```

- **5.25**
  What letters are interrogated by the *binary search* `BinSearch` in question **II.1.**b) in the next page if it is applied to the following list when searching for the value `J`? What about searching for the value `Z`?

```
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O
```

- **5.53**
  The following algorithm is designed to compute the product of two nonnegative integers `X` and `Y` by accumulating the sum of `X` copies of `Y` ; that is, `3` times `4` is computed by accumulating the sum of `4`, `4`, and `4`. Is the algorithm correct? If yes, explain; otherwise correct it.

```
Product ← 0;
Count ← 0;
repeat {
    Product ← Product + Y,;
    Count ← Count + 1;
} until (Count = X)
```

- **5.54**
  The following algorithm is designed to report which of the positive integers `X` and `Y` is larger. Is the algorithm correct? If yes, explain; otherwise correct it.

```
Difference ← X - Y;
if (Difference > 0)
then {
    print “X is bigger than Y”;
}
else {
    print “Y is bigger than X”;
}
```

- **5.57**
  Based on the *preconditions* that `X` and `Y` are assigned nonnegative integers, identify the *loop invariant* for the following **`while`** structure which, when combined with the *stop condition*, implies that the value associated with `Z` upon loop termination must be `X - Y`.

```
Z ← X;
J ← 0;
while (J < Y) do {
    Z ← Z - 1;
    J ← J + 1;
}
```

II Answer the following questions:

1. Determine the class of each of the following algorithms:

a)

```
procedure SeqSearch(L,V)
if (isEmpty(L))
then {
    print "Not Found.";
}
else {
    T ← first(L);
    while (V > T and T isNot last(L)) do {
        T ← next(L);
    }
    if (V = T)
    then {
        print "Found!";
    }
    else {
        print "Not Found.";
    }
}
```

b)

```
procedure BinSearch(L,V)
if (isEmpty(L))
then {
    print "Not Found.";
}
else {
    T ← middle(L);
    if (V = T)
    then {
        print "Found!";
    }
    else {
        if (V < T)
        then {
            BinSearch(before(T,L),V);
        }
        else {
            BinSearch(after(T,L),V);
        }
    }
}
```

2. Identify the *preconditions* for the sequential search SeqSearch presented in the previous question. Establish a *loop invariant* for the while structure which, when combined with the *stop condition*, implies that the algorithm will report "Found" or "Not Found" correctly upon termination of the loop. Give an argument showing that the while loop does in fact terminate.

3. Are the following two algorithms equivalent? Explain.

a)

```
procedure RecFact(N)
if (N ≤ 1)
then {
    return 1;
}
else {
    return N * RecFact(N-1)
}
```

b)

```
procedure NonRecFact(N)
F ← 1;
Count ← 1;
while (Count ≤ N) do {
    F ← F * Count;
    Count ← Count + 1;
}
return F;
```

4. Arrange the following algorithm classes in order of decreasing performance. Give an example for each class.

$\Theta(\log n), \Theta(n\log n), \Theta(n), \Theta(n^2), \Theta(1)$