1. (20 points) A network topology specifies how computers, printers, and other devices are connected over a network. Figure 1 below illustrates three common topologies of networks: the ring, the star, and the fully connected mesh.

![Figure 1: Topologies](image)

You are given a boolean matrix \(A[0..n-1, 0..n-1]\), where \(n > 3\), which is supposed to be the adjacency matrix of a graph modeling a network with one of these topologies, if any, the matrix represents. Design a brute-force algorithm (express it in pseudocode) for this task and indicate its time efficiency class.

2. (25 points) Consider the following algorithm:

```
Algorithm 1 CountingSort(A[1..n], k)
1: {Sort the elements in A[1..n] in nondecreasing order}
2: {Input: An array A[1..n] of nonnegative integers and the value (k) of the highest element in A[1..n]}
3: {Output: An array B[1..n] containing the elements of A[1..n] sorted in nondecreasing order}
4: for i ← 0 to k do
5:   C[i] ← 0
6: end for
7: for i ← 1 to n do
8:   C[A[i]] ← C[A[i]] + 1
9: end for
10: for i ← 1 to k do
11:   C[i] ← C[i] + C[i - 1]
12: end for
13: for i ← n downto 1 do
14:   B[C[A[i]]] ← A[i]
15:   C[A[i]] ← C[A[i]] - 1
16: end for
```

(a) Apply `CountingSort` algorithm to sort the list \(\langle 2, 5, 3, 0, 2, 3, 0, 3 \rangle\). Specifically, show the states of the array \(C\) just after line no. 9 and just after line no. 12. And also show the state of the array \(B\) after each iteration of the for loop on line no. 13.

(b) Do we need to consider the worst-case, the average-case, and the best-case while analyzing the algorithm? Explain your answer. What is the time efficiency class of this algorithm?
(c) Is it a stable sorting algorithm? Explain your answer. Give one situation where we may prefer a stable sorting algorithm over an unstable one.

(d) Is it an in-place sorting algorithm? Explain your answer.

3. (15 points) Give an example of a text of length \( n \) and a pattern of length \( m \) (> 1) that constitutes a worst-case input for the brute-force string matching algorithm (given below). Exactly how many character comparisons will be made for such input?

```
Algorithm 2 BruteForceStringMatching(T[0..n - 1], P[0..m - 1])

1: {Implements brute-force string matching}
2: {Input: An array T[0..n - 1] of n characters representing a text and an array P[0..m - 1] of m characters representing a pattern}
3: {Output: The index of the first character in the text that starts a matching substring or -1 if the search is unsuccessful}
4: for \( i \leftarrow 0 \) to \( n - m \) do
5:   \( j \leftarrow 0 \)
6:   while \( j < m \) and \( P[j] = T[i + j] \) do
7:     \( j \leftarrow j + 1 \)
8:   end while
9:   if \( j = m \) then
10:      return \( i \)
11:   end if
12: end for
13: return -1
```

4. (10 points) How can pseudorandom numbers be useful in the empirical analysis of an algorithm?

5. (10 points) Given \( f(n) = n2^n \) and \( g(n) = 3^n \), which one do you think is true: \( f(n) \in O(g(n)) \), \( f(n) \in \Omega(g(n)) \), or \( f(n) \in \Theta(g(n)) \). Prove your answer.

6. (20 points) Solve the following recurrence relation (assume that \( n \) is a power of 2):

\[
T(n) = \begin{cases} 
1 & \text{when } n = 1, \\
2T \left( \frac{n}{2} \right) + 6n - 1 & \text{when } n \geq 2
\end{cases}
\]

Given \( T(n) \in O(g(n)) \), \( g(n) = \) ?

1 Useful formulas

1. \( \sum_{i=l}^{u} 1 = u - l + 1 \)
2. \( \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \)
3. \( \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \)
4. \( \sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} (a \neq 1) \)
5. \( \lg n = \frac{\ln n}{\ln 2} \)
6. \( \frac{d}{dn} n^m = mn^{m-1} (m \neq 1) \)
7. \( \frac{d}{dn} \ln n = \frac{1}{n} \)
8. \( \frac{d}{dn} a^n = (\ln a)a^n \)