Note: YOU MUST DO THE HOMEWORK BY YOURSELF. If you have difficulties in solving a question you may discuss it with friends, BUT you MUST phrase, write and formulate the answers by yourself, after you understand the solution. The language of the solution must be entirely your own. If you got some idea from a friend or a certain paper please cite her/him/it and give her/him/it credit. It will not harm you.

Notice 2: Please type (Word, or LaTeX, etc.) and submit your homework as pdf file to the email address that was given.

Notice 3: Put your name and id number on each page of the solution.

1. (a) Prove that in any run of the distributed MST algorithm of Gallager Hublet and Spira, (that was described in class) in any connected network the set of cores, and the level at which each core is a core, in the run is the same (regardless of the scheduler).

(b) Give an abstract example network on which the MST algorithm completes in one phase.

(c) Is it true that when the Gallager Hublet and Spira MST algorithm runs on a network whose diameter is \(D\) then it running time is \(O(D \log n)\) time?

2. Based on the centralized Depth First Search (DFS) procedure, describe a distributed DFS traversal algorithm for an asynchronous network. In a traversal algorithm a central node, called root, initiates a token which has to visit all the nodes of the network, one at a time. Argue that in order to visit all the nodes the token must traverse all the links of the network. (links might be traversed more than one time)

(a) (Half a page) Verify that your algorithm uses \(O(|E|)\) messages and \(O(|E|)\) time, where \(|E|\) is the total number of links in the network.

(b) (At most one page) Modify your algorithm to reduce its time complexity to \(O(n)\), where \(n\) is the total number of nodes in the network. (Hint: in the modification there must be times at which there are more than one message in transit in the network.)

3. Draw and describe in detail the following protocol complexes:

(a) 2 rounds, 2 processors, 1-Mobile, Omission fault. Invent a simple descriptive method to describe the history of a processor in each node of the complex.

(b) 2 rounds, 2 processors, 1-Stationary, Omission fault.

(c) What conclusions did we draw from the different structure of these two complexes?
4. **Phase-king protocol of Berman Garay:** Prove the correctness of the following 3\((t + 1)\) rounds synchronous consensus algorithm if \(n > 3t\) (Byzantine Agreement2 in Figure 1) (first prove the 4 claims below).

It uses messages of constant size (2 bits), and runs in \(t + 1\) phases, each consisting of three exchange rounds. Each processor has a local variable \(V\), and integer arrays \(C\) and \(D\). Value 2 represents ‘undecided’, whereas 1 is used as the default value.

**Byzantine Agreement2:** \(3t + 1\) rounds protocol when \(n > 3t\). Code for process \(i\).

\[
V := v_i; \quad // (* i 's initial value *)
\]

\[
\text{for } m := 1 \text{ to } t + 1 \begin{cases}
\text{(* Exchange 1 *)} \\
\text{send}(V); \\
V := 2; \\
\text{for } k := 0 \text{ to } 1 \begin{cases}
\text{(* Exchange 2 *)} \\
C(k) := \text{the number of received } k \text{'s;}
\text{if } C(k) \geq n - t \text{ then } V := k
\end{cases}
\end{cases}
\]

\[
\text{send}(V); \\
\text{for } k := 2 \text{ downto } 0 \begin{cases}
\text{(* Exchange 3 *)} \\
D(k) := \text{the number of received } k \text{'s;}
\text{if } D(k) > t \text{ then } V := k
\end{cases}
\]

\[
\text{if } m = i \text{ then} \\
\text{send}(V); \\
\text{if } V = 2 \text{ or } D(V) < n - t \text{ then} \\
V := \text{MIN}(1, \text{received message});
\]

\]

(a) **Proof:**

To prove the algorithm first provide a proof of each of the following three:

i. At the end of Exchange 1 there exists \(v \in \{0, 1\}\) such that, for all correct processors, the value of \(V\) is \(v\) or 2.

ii. At the end of Exchange 2, for all correct processors the value of \(V\) is \(v\) or 2 (same \(v\) as above).

iii. If at the beginning of a given phase the value of \(V\) is equal to \(v\) for all correct processors, the same is true at the end of this phase. (Persistency of agreement)

Next provide a detailed proof of the following (i.e., add details to the following proof):
Let $g$ be the smallest number of a correct processor. At the end of Exchange 3 of phase $g$ agreement is reached, because either

i. all correct processors accept the phase king’s message (and thus reach consensus), or else

ii. some correct processors ignore this message because they find $D(V) < n - t$ false. But then $D(V) > t$ for all correct processors, and so their values of $V$ are equal after Exchange 3 whether they accept the king’s message or not.

Using the 4 claims you proved above, complete the proof of the algorithm correctness.

(b) Can you identify parts of this algorithm that correspond to the algorithm described in class (using the shared-memory and commit adopt constructs)? Compare the two algorithms, most importantly in their structure and then in complexities (ignoring constants).

(c) What will go wrong in the above Phase-King Protocol $(n > 3f)$, if in Exchange 2 the line

\[
\text{for } k := 2 \text{ downto } 0 \text{ do begin}
\]

will be replaced with the line:

\[
\text{for } k := 0 \text{ upto } 2 \text{ do begin}
\]

Give an example (a Scenario) in which the algorithm would fail.

5. Give an agreement (consensus) protocol for $n$ processes in a synchronous system (clique) with fail-stop failures such that, (validity) if all non faulty processes have the same initial value $v$ then all correct (non faulty) processes decide $v$. The input values are binary. Processes know $n$. In addition the maximum number of faulty processes is $t < N/2$, and let $g$ be the actual number of faulty processes ($g \leq f$). The big-O number of rounds of your protocol should be a very small function (e.g., linear) of $g$. 