Titas Gas Transmission and Distribution Company Limited (TGTDCL)
Assistant Engineer (CSE), Exam Taker: BUET
Exam Time: 1 hour, 09-10 AM Date: 24/05/2024 Non Dept: 20, Dept: 40
Nusrat & Raihan Tarek

1. a) **Write recursive function for the given scenario:** **[2]**

```
for (int i=1,i<n;i++){
        For ( int j=0 ; j<i ; j ++){
                For ( int k =0; k<i ; k++ )
                x= x+1;
}}};
```

```
#include <iostream>
void innerLoop(int i, int k, int &count) {
   if (k < i) {
      count++;      // Equivalent to X = X + 1
      innerLoop(i, k + 1, count);
   }
}
void middleLoop(int i, int j, int &count) {
   if (j < i) {
      innerLoop(i, 0, count);
      middleLoop(i, j + 1, count);
   }
}
void outerLoop(int i, int n, int &count) {
   if (i < n) {
      middleLoop(i, 0, count);
      outerLoop(i + 1, n, count);
   }
}
int main() {
   int n = 5; // Example value for n
   int count = 0; // Initialize count
   outerLoop(1, n, count);
   cout << "Number of times x = x + 1 is executed: " << count <<endl;
   return 0;
}
```

b)	**If a complete binary tree is a proper binary tree, then prove that number of leaf node is one more than internal nodes. [3]**

Total Nodes (N): The total nodes in the tree are the sum of internal nodes and leaf nodes.
N = I + L
Relationship in Full Binary Tree: In a full binary tree, the number of total nodes is given by:
N = 2I + 1
This is because each internal node contributes exactly two children, and there is one additional internal node.
Proof

1. Starting with the relationship between the total number of nodes (N) and the internal nodes (I):
   $N=2I+1$
2. From the total nodes equation, we also have:
   $N=I+L$
3. Equating the two expressions for N:
   $I+L=2I+1$
4. Solving for L:
   $L=2I+1-I$
   $L=I+1$

Conclusion
The number of leaf nodes (L) in a complete binary tree that is full is exactly one more than the number of internal nodes (I):
$L=I+1L=I+1$
Thus, we have proven that in a complete binary tree, the number of child nodes (leaf nodes) is one more than the number of internal nodes.

c)	**Problem A can be reduced to problem B in polynomial time, which problem is harder and why? [3]**

Given this reduction, the relative hardness of the problems can be assessed:
Hardness Comparison
Problem B is at least as hard as Problem A: If problem A can be reduced to problem B in polynomial time, solving problem B must be at least as hard as solving problem A. This is because, if we had an efficient (polynomial-time) solution for problem B, we could use the reduction to efficiently solve problem A as well.
Why Problem B is Considered Harder
Efficiency of Reduction: Since the reduction from A to B is polynomial-time, it doesn't significantly increase the overall complexity. Therefore, if we can solve B efficiently (in polynomial time), we can also solve A efficiently by first transforming it into an instance of B and then solving B.

Implications for Complexity Classes: If problem A is in a lower complexity class (e.g., P) and problem B is in a higher complexity class (e.g., NP-complete), then the reduction implies that problem B is at least as hard as problem A. Specifically, if problem A is NP-complete and we can reduce A to B in polynomial time, then B must be NP-hard (and possibly NP-complete if it's also in NP).

Example

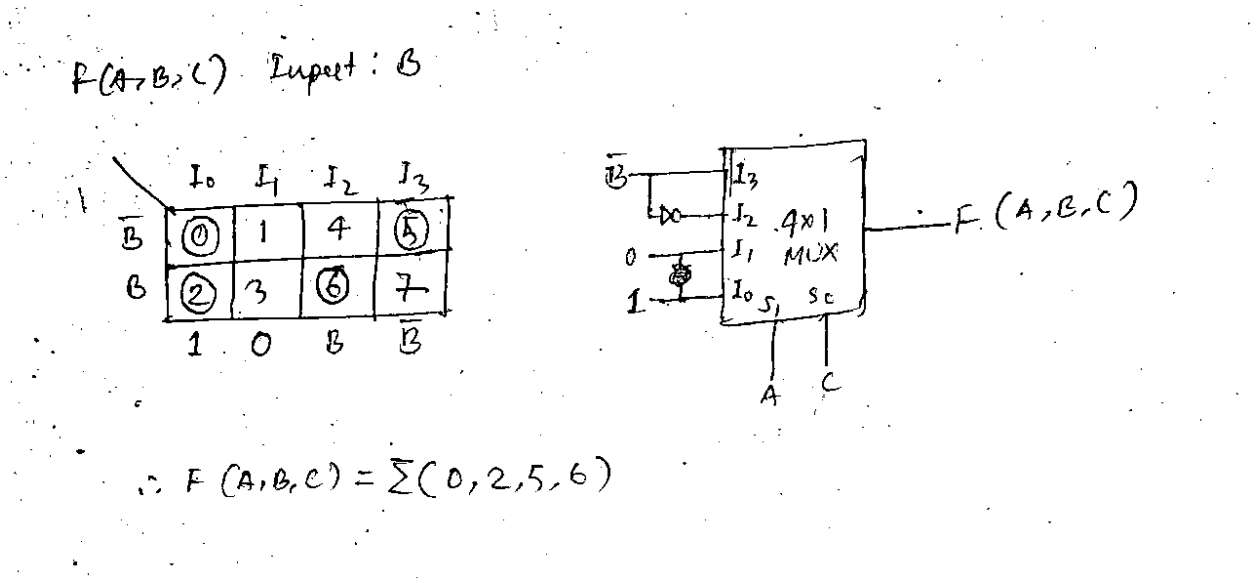
Consider problems A and B where:

Problem A: 3-SAT (which is NP-complete)

Problem B: Some other problem

If 3-SAT (problem A) can be reduced to problem B in polynomial time, this suggests that problem B is at least as hard as 3-SAT

2. a)	**A 4*1 MUX is given, write down its expression in SOM form. [4]**



According to Raihan Tarek Question collection and answer

b)	**Write down which is related to CISC or RISC from given description [2]**

Easy to pipe line-	RISC
More Addressing mode	CISC
fixed instruction format.	RISC
More memory efficient	CISC

c)	**What is page fault? [2]**

A page fault is an event that occurs in a computer system's memory management when a program tries to access a specific piece of data (a page) that is not currently loaded in physical memory (RAM). This means the data is either:

1. **Not yet loaded:** The program might be referencing a page from disk for the first time.
2. **Swapped out:** The page was previously in RAM but was removed to make space for other data and needs to be retrieved from secondary storage (like a hard disk).

3. a)	**Briefly explain Circular wait in dead lock. [2]**

Circular wait, also known as a circular dependency, is a necessary condition for a deadlock to occur in a computer system. It's a situation where multiple processes are waiting for resources that are currently held by other processes in the system, creating a loop of dependencies.
Here's a breakdown of circular wait and its role in deadlock:

Circular Wait:

- A set of processes are each waiting for a resource that is held by another process in the sequence.
- This creates a circular chain of dependencies, where each process is blocked because it needs a resource held by the next process in the chain.

3 (b)	**What is the appropriate pairing of items in the two columns listing various activities encountered in a software life cycle?**

P.	Requirements Capture	1.	Module Development and Integration
Q.	Design	2.	Domain Analysis
R.	Implementation	3.	Structural and Behavioral Modeling
S.	Maintenance	4.	Performance

P - 2, Q - 3, R - 1, S – 4;

c)	**How to prevent XSS And CSRF attack? [2]**

Prevention XSS:

- **Prevention:**
  - **Input Validation:** Sanitize all user input (form data, URL parameters, etc.) to remove potentially harmful scripts.
  - **Output Encoding:** Encode special characters in user input before displaying it on the website to prevent them from being interpreted as code.
  - **Content Security Policy (CSP):** Define a whitelist of allowed sources for scripts to run on your website. This restricts the execution of any unauthorized scripts.

Prevention CSRF: Cross-Site Request Forgery (CSRF):

- **Prevention:**
  - **CSRF Tokens:** Embed a unique, random token in every form on your website. Include this token in the request body or a hidden form field. The server verifies the token to ensure the request originated from your website and not a forged link.
  - **SameSite Cookie Attribute:** Set the SameSite attribute of cookies to Lax or Strict to restrict how cookies are sent in cross-site requests.

4. a)	**A log file keeps record of the updates, how to get the last line of the log file? [2]**

tail -n 1 /path/to/logfile.log

b)	**traceroute to reach marketplace.com where the traceroute sequence is given bellow,** **1. gateway.titasgas.gov.bd ip time** **2. backup.titasgas.gov.bd ip time** **.** **.** **.** **.** **8. global.com ip time** **9. * * *** **10. * * *** **11. * * *** **12. marketplace.com** **How many hops are required to reach 12? what will be in line 9 to 11? Explain. [4]**

Hops Required: It takes **12 hops** to reach marketplace.com.

Lines 9-11: Timeouts or Packet Filtering

c)	**How Man In The Middle attack affect on DHCP server? [2]**

In a MITM attack on DHCP, an attacker inserts themselves into the communication between a DHCP server and clients. The attacker can achieve this by spoofing DHCP messages, either acting as a rogue DHCP server or intercepting and modifying DHCP traffic. Here are a few common scenarios:

1. **Rogue DHCP Server**: The attacker sets up a rogue DHCP server on the network. When a client requests an IP address, the rogue server responds faster than the legitimate server, providing the client with malicious configuration settings.
2. **DHCP Spoofing**: The attacker intercepts DHCP requests and responses, modifying them before they reach the intended recipient. This can include altering IP addresses, DNS servers, gateways, etc

5. a)	**Write query to find out supliers name who supplies 'wheel' at least cost. [4]** **Suppliers (sid, sname, address)** **Parts (pid, pname, color)** **Catalog (sid, pid, cost)**

```
SELECT S.sname
FROM Suppliers S
JOIN Catalog C ON S.sid = C.sid
JOIN Parts P ON C.pid = P.pid
WHERE P.pname = 'wheel' AND C.cost = (
   SELECT MIN(C2.cost)
   FROM Catalog C2
   JOIN Parts P2 ON C2.pid = P2.pid
   WHERE P2.pname = 'wheel'
```

b)	**Write a program to findout an expression is ended correctly or not. [4]** **Input: ((a+b)+(a-b) Output: Matched** **Input: (a+b)+a-c) Output : Not matched**

```
#include <iostream>
#include <string>
using namespace std;
bool isMatched(const string& expression) {
```

```
    int balance = 0;

    for (char ch : expression) {
        if (ch == '(') {
            balance++;
        } else if (ch == ')') {
            balance--;
        }
        if (balance < 0) {
            return false;
        }
    }
    return balance == 0;
}

int main() {
    string expr1 = "((a+b)+(a-b)";
    string expr2 = "(a+b)+a-c)";

cout << "Input: " << expr1 << "  Output: " << (isMatched(expr1)
? "Matched" : "Not matched") << endl;
cout << "Input: " << expr2 << "  Output: " << (isMatched(expr2)
? "Matched" : "Not matched") << endl;
    return 0;
}
```