**THE UNIVERSITY OF DODOMA**

**COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Assignment Title:** 01

**Course Name:** COMPUTATIONAL THEORY

**Course Code:** CP 214

**Group Number:** 4

| S/N | NAME | REGISTRATION NUMBER |
|---|---|---|
| 1 | Victoria Zacharia | T24-03-17605 |
| 2 | Daniel Elias | T24-03-10364 |
| 3 | Witness Joshua | T24-03-16219 |
| 4 | Saleh Nassor Said | T24-03-20761 |
| 5 | Praise Mashiku | T24-03-10358 |
| 6 | George Mwaluko | T24-03-20754 |
| 7 | Dominic Nazareth | T24-03-15493 |
| 8 | Collin Agaba | T24-03-13894 |
| 9 | Joshua Rodgers | T24-03-15836 |

## GROUP 4 RESPONSE:

**Qn1:** *Major areas of the theory of computation*

i. **Automata:** It deals with definition and properties of mathematical models.
ii. **Computability:** It focuses on categorizing problems into solvable and non-solvable.
iii. **Complexity:** It studies hard problems against easy problems.

**Qn2**: *Importance of the theory of computation*

The study of computation is important because it provides or explains a foundation on how to solve mathematical problems in computer science and how to formulate algorithms.

**Qn3:** *Strategies for handling computationally hard problems*

i. **Modify or simplify** a problem to make it solvable.
ii. Use **approximation algorithms** that give near optimal solution.
iii. Use **randomized techniques** that work well in practice even if not perfect.

**Qn4:** *Relationship between complexity theory and cryptography*

Complexity Theory falls in line with cryptographic problems, e.g., in cryptography focus on hard computational problems rather than easy ones in order to ensure secrecy. In cryptography, for instance, secret codes should be hard to break without the secret key or passwords. Complexity Theory has pointed cryptographers in the direction of computationally hard problems around which they have designed revolutionary new codes.

**Q.5:** *Purpose of classifying problems according to difficulty*

i. To determine which problems are efficiently solvable.
ii. To compare using **reductions**.
iii. To know which solution strategies to use (**exact, approximate, heuristic**).
iv. To understand the **limitations of computation**.

**Q.6:** *Why are approximations sometimes acceptable in solving hard problems*

Approximations are sometimes acceptable when solving problems because many real-world applications do not require perfectly exact solutions. Approximation algorithms can produce answers that are close to optimal while running much faster and using fewer resources. In contrast, finding an exact solution may take too long or demand excessive computational resources, making approximation the more practical choice.

**Q.7**: *How sequences differ from sets*

A sequence is ordered and allows repetitions while a set is unordered and does not allow duplicates.

**Q.8**: *Given $A = \{1, 2\}$ and $B = \{x, y\}$*

$A \times B = \{(1, x), (1, y), (2, x), (2, y)\}$

**Q.9:** *All elements of: $A \times B \times A$*

$A \times B \times A = \{(1, x, 1), (1, x, 2), (1, y, 1), (1, y, 2), (2, x, 1), (2, x, 2), (2, y, 1), (2, y, 2)\}$.

**Q.10:** *Number of elements in $A \times B$*

$|A \times B| = a \times b$, because each element of A can pair with each element of B.

**Q.11:** *Given C is a set with c elements, how many elements are in the power set of C*

$|P| = 2^c$, because each element of C can either be included or not in subsets.

**Q.12:** *Given functions and their response tables*

a) $f(2) = 7$

b) Range of $f = \{y: 6 \leq y \leq 7\}$ and Domain of $f = \{x: 1 \leq x \leq 5\}$

c) $g(2, 5) = 6$

d) Domain of $g = \{(x, y): 1 \leq x \leq 5 \text{ and } 6 \leq y \leq 10\}$

Range of $g = \{y: 6 \leq y \leq 10\}$

e) $g(4, f(4)) =$

Solution

$f(4) = ?$

So, $g(x, f(4)) = g(4, 7) = 8$

$g(4, f(4)) = 8$

**Q.13**: _Describe how a function can be represented using a table_.

- For **unary functions**, a two-column table showing each input and its corresponding output.
- For **binary functions**, a matrix where rows represent one input, columns represent the second, and each cell contains the output.

**Q.14:** _Given graph G =({1,2,3},{(1,2), (2,3)}), list:_

a) ***Degrees of all nodes***

Node 1 has **degree of 1**

Node 2 has **degree of 2**

Node 3 has **degree of 1**

b) ***All paths from 1 to 3***

(1, 2, 3)

**Q.15:** _Is "cad" a substring of "abracadabra"? Justify_.

Yes, because "cat" appears inside "abracadabra" starting at position 5.