Bangladesh Army International University of Science & Technology
Department of Computer Science and Engineering

<u>Lab Report</u>

Lab Report No	**04**
Lab Report Name	**Implementation of Dijkstra's Shortest Path Algorithm**
Course Title	**Computer Algorithms & Complexity Sessional**
Course Code	**CSE 222**
Name	Rayhan Alam Towhid
ID	1119007

Level	2	**Term**	II	**Section**	A	**Group**	G1
Date of Submission	02-11-2025			**Session**	Fall-2025		

Marking Rubric:

Problem Understanding & Report Clarity (3)	Implementation (5)	Results & Analysis (2)	Total (10)

Key Learnings:

Dijkstra's Algorithm teaches how to find the shortest path from a source to all other vertices in a weighted graph. You learn about greedy strategy, priority of vertices, and updating distances efficiently.

Code Implementation:

By Rayhan6244, contest: Codeforces Round 940 (Div. 2) and CodeCraft-23, problem: (A) Stickogon, **Accepted**, #, Copy

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin >> t;

    while (t--) {
        int n;
        cin >> n;

        int freq[101] = {0};

        for (int i = 0; i < n; i++) {
            int x;
            cin >> x;
            freq[x]++;
        }

        int ans = 0;
        for (int i = 1; i <= 100; i++) {
            ans += freq[i] / 3;
        }

        cout << ans << endl;
    }

    return 0;
}
```

Sample Input - Output:

Test: #1, time: 31 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input
```
4
1
1
2
1 1
6
2 2 3 3 3 3
9
4 2 2 2 2 4 2 4 4
```

Output
```
0
0
1
2
```

Answer
```
0
0
1
2
```

Checker Log
```
ok 4 number(s): "0 0 1 2"
```

Test: #2, time: 31 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input
```
100
2
1 1
4
1 1 1 1
7
1 1 1 1 1 1 1
```

Result Analysis / Discussion:

Dijkstra's Algorithm starts from a source vertex, repeatedly selects the vertex with the smallest known distance, and updates the distances of its neighbors. It efficiently finds the shortest paths in graphs with non-negative edge weights using a priority queue or simple array.