

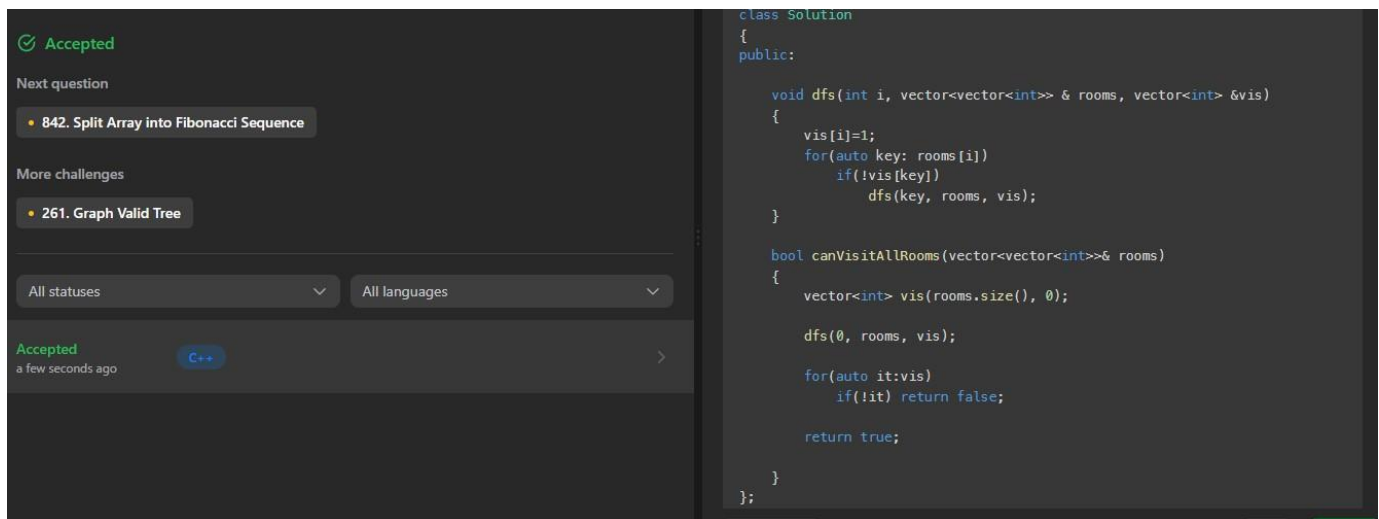
WORKSHEET 7

Student Name: Harsh Chauhan

UID: 21BCS10053

Subject Name: IT Skills (DSA)

Question 1. KEYS AND ROOMS



The screenshot displays a coding interface with a dark theme. On the left sidebar, a green checkmark and the word "Accepted" are visible at the top. Below this, the text "Next question" is followed by a button labeled "842. Split Array into Fibonacci Sequence". Under "More challenges", there is a button labeled "261. Graph Valid Tree". Further down, there are two dropdown menus: "All statuses" and "All languages". At the bottom of the sidebar, another "Accepted" status is shown with the text "a few seconds ago" and a "C++" language selector. The main area on the right shows a C++ code snippet for a class named "Solution". The code implements a Depth-First Search (DFS) algorithm to check if all rooms can be visited. It includes a recursive function "dfs" and a public function "canVisitAllRooms".

```
class Solution
{
public:
    void dfs(int i, vector<vector<int>> & rooms, vector<int> &vis)
    {
        vis[i]=1;
        for(auto key: rooms[i])
            if(!vis[key])
                dfs(key, rooms, vis);
    }

    bool canVisitAllRooms(vector<vector<int>>& rooms)
    {
        vector<int> vis(rooms.size(), 0);

        dfs(0, rooms, vis);

        for(auto it:vis)
            if(!it) return false;

        return true;
    }
};
```

Question 2. HIDDEN COLORED GRAPH

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  bool query(int v) {
4      cout << "? " << v << endl;
5      char c;
6      cin >> c;
7      return c == 'B';
8  }
9
10 int main() {
11     ios::sync_with_stdio(false);
12     cin.tie(0);
13     int n;
14     cin >> n;
15     vector<vector<bool>> q(n + 1);
16     vector<int> ve;
17     ve.push_back(1);
18     for(int i = 1; i <= n; i++) {
19         ve.push_back(i);
20     }
21     for(int k : ve) {
22         for(int i = k; i <= n; i++) {
23             q[i].push_back(query(i));
24         }
25     }
26     vector<vector<bool>> adj(n + 1, vector<bool>(n + 1));
27     for(int i = 2; i <= n; i++) {
28         for(int j = 1; j < i; j++) {
29             adj[i][j] = adj[j][i] = (q[i][j - 1] ^ q[i][j + 1]);
30         }
31     }
32     cout << "!\n";
33     for(int i = 1; i <= n; i++) {
34         for(int j = 1; j <= n; j++) {
35             cout << adj[i][j];
36         }
37         cout << '\n';
38     }
39     cout << flush;
40 }

```

SOLUTION:

Status: ✔ Correct Answer

Submission ID: [85053533](#)

Score:	Time:	Memory:
1	0.02s	5.4M

Question 4.

Question 3. WINTER

```
1 #include<bits/stdc++.h>
2
3 #define int long long int
4 #define F first
5 #define S second
6 #define pb push_back
7 #define que_max priority_queue<int>
8 #define que_min priority_queue<int,vector<int>,greater<int>>;
9 #define endl "\n"
10 using namespace std;
11
12 int32_t main()
13 {
14     #ifndef ONLINE_JUDGE
15         freopen("input.txt","r",stdin);
16         freopen("output.txt","w",stdout);
17     #endif
18
19     int n,m,q1;
20     cin>>n>>m>>q1;
21     vector<vector<int>>vec(n+1);
22     for(int i=0;i<m;i++)
23     {
24         int x,y;
25         cin>>x>>y;
26         vec[x].push_back(y);
27         vec[y].push_back(x);
28     }
29
30     vector<bool>visited(n+1,false);
31     queue<int>q;
32     vector<bool>frozen(n+1,false);
33
34     while(q1-->0)
35     {
36
37         int query,type;
38         cin>>type>>query;
39         if(type==1)
40         {
```

```
41             if(frozen[query])continue;
42             frozen[query]=true;
43             // if(visited[query]==false)
44             // {
45             //     visited[query]=true;
46             //     q.push(query);
47             // }
48         }else if(type==2)
49         {
50             while(q.size()!=0 && query!=0 )
51             {
52                 int sz=q.size();
53                 while(sz-->0){
54                     int tp=q.front();
55                     visited[tp]=true;
56                     q.pop();
57
58                     for(auto nbr:vec[tp])
59                     {
60
61                         if(!visited[nbr]){
62                             if(frozen[nbr]) continue;
63                             frozen[nbr]=true;
64                             q.push(nbr);
65                         }
66                     }
67                     query--;
68                 }
69             }else
70             {
71                 if(frozen[query])
72                 {
73                     cout<<"Yes"<<endl;
74                 }else
75                 {
76                     cout<<"No"<<endl;
77                 }
78             }
79         }
80     }
81     return 0;
82 }
```

SOLUTION:

Query	Type	Query	Result
1	1	1	Yes
2	1	2	Yes
3	1	3	Yes
4	1	4	Yes
5	1	5	Yes
6	1	6	Yes
7	1	7	Yes
8	1	8	Yes
9	1	9	Yes
10	1	10	Yes
11	1	11	Yes
12	1	12	Yes
13	1	13	Yes
14	1	14	Yes
15	1	15	Yes
16	1	16	Yes
17	1	17	Yes
18	1	18	Yes
19	1	19	Yes
20	1	20	Yes
21	1	21	Yes
22	1	22	Yes
23	1	23	Yes
24	1	24	Yes
25	1	25	Yes
26	1	26	Yes
27	1	27	Yes
28	1	28	Yes
29	1	29	Yes
30	1	30	Yes
31	1	31	Yes
32	1	32	Yes
33	1	33	Yes
34	1	34	Yes
35	1	35	Yes
36	1	36	Yes
37	1	37	Yes
38	1	38	Yes
39	1	39	Yes
40	1	40	Yes
41	1	41	Yes
42	1	42	Yes
43	1	43	Yes
44	1	44	Yes
45	1	45	Yes
46	1	46	Yes
47	1	47	Yes
48	1	48	Yes
49	1	49	Yes
50	1	50	Yes
51	1	51	Yes
52	1	52	Yes
53	1	53	Yes
54	1	54	Yes
55	1	55	Yes
56	1	56	Yes
57	1	57	Yes
58	1	58	Yes
59	1	59	Yes
60	1	60	Yes
61	1	61	Yes
62	1	62	Yes
63	1	63	Yes
64	1	64	Yes
65	1	65	Yes
66	1	66	Yes
67	1	67	Yes
68	1	68	Yes
69	1	69	Yes
70	1	70	Yes
71	1	71	Yes
72	1	72	Yes
73	1	73	Yes
74	1	74	Yes
75	1	75	Yes
76	1	76	Yes
77	1	77	Yes
78	1	78	Yes
79	1	79	Yes
80	1	80	Yes
81	1	81	Yes
82	1	82	Yes
83	1	83	Yes
84	1	84	Yes
85	1	85	Yes
86	1	86	Yes
87	1	87	Yes
88	1	88	Yes
89	1	89	Yes
90	1	90	Yes
91	1	91	Yes
92	1	92	Yes
93	1	93	Yes
94	1	94	Yes
95	1	95	Yes
96	1	96	Yes
97	1	97	Yes
98	1	98	Yes
99	1	99	Yes
100	1	100	Yes

MINIMAL TRAVEL TIME

```

1 #include <bits/stdc++.h>
2
3 #define llint long long int
4 using namespace std;
5
6 void run()
7 {
8     // Insert code here
9     int n, m, s, k;
10    cin >> n >> m >> s >> k;
11
12    vector<vector<int>> graph(n+1);
13
14    for(int i = 0; i < m; ++i){
15        int u, v;
16        cin >> u >> v;
17        graph[u].push_back(v);
18        graph[v].push_back(u);
19    }
20
21    std::vector<int> count(n+1);
22    for (int i = 0; i < s; ++i){
23        int val;
24        cin >> val;
25        count[val]++;
26    }
27    vector<bool> vis(n+1);
28    queue<int> q;
29
30    q.push(0);
31    vis[0] = true;
32
33    llint res = 0, curr = 0;
34
35    while(!q.empty() && k > 0){
36        int size = q.size();
37        for(int i = 0; i < size; ++i){

```

```

38            int node = q.front();
39            q.pop();
40            for(auto adj : graph[node]){
41                if(!vis[adj]){
42                    vis[adj] = true;
43                    q.push(adj);
44                }
45            }
46            int val = min(k, count[node]);
47            res += 2*curr*val;
48            k -= val;
49        }
50        curr++;
51    }
52    cout << res << "\n";
53 }
54
55 int main()
56 {
57     std::ios_base::sync_with_stdio(false);
58     std::cin.tie(NULL);
59
60     int t = 1;
61     std::cin >> t;
62     while (t--){
63         run();
64     }
65     return 0;
66 }

```

SOLUTION:

Status: ✓ Correct Answer		Submission ID: 85054109
Score: 100	Time: 0.15s	Memory: 10.3M
Sub-Task	Task #	Result (time)
1	0	AC (0.153760)
1	1	AC (0.039809)
1	2	AC (0.087936)
1	3	AC (0.086389)
Subtask Score: 100.00%		Result - AC
Total Score = 100.00%		



Question 6.

Question 7.

CHEF AND REVERSING

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int N = 1e5+10;
4  const int infi=1e9+10;
5  vector<pair<int,int>>g[N];
6  vector<int>level(N,infi);
7  int n,m;
8  void bfs(){
9      level[1]=0;
10     deque<int> dq;
11     dq.push_back(1);
12     while(!dq.empty()){
13         int cur_v= dq.front();
14         dq.pop_front();
15         for(auto child:g[cur_v]){
16             int child = child.first;
17             int wt = child.second;
18             if(level[cur_v]+wt < level[child]){
19                 level[child] = level[cur_v] + wt;
20                 if(wt==1) dq.push_back(child);
21                 else dq.push_front(child);
22             }
23         }
24     }
25     if(level[n]==infi) cout<<-1 ;
26     else cout<<level[n];
27 }
28
29 int main() {
30
31     cin>>n>>m;
32     for(int i=0;i<m;i++){
33         int x,y;
34         cin>>x>>y;
35         if(x==y)continue;
36         g[x].push_back({y,0});
37         g[y].push_back({x,1});
38     }
39     bfs();
40
41     return 0;
42 }

```

SOLUTION:

Status: ✔ Correct Answer

Submission ID: [85054172](#)

Time:
0.05s

Memory:
8.9M

Question 8.

CHEF AND EDGE FLIPPING

```
Language: C++14

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define N 1010
5
6  int n, m, a[N], b[N];
7  bool col[N], s[N][N];
8
9  bool check(int u) {
10     for (int i = 1; i <= n; i++) col[i] = 0; col[u] = 1;
11     for (int i = 1; i <= n; i++) if (i != u) s[i][u] = 1, s[u][i] = 0;
12     for (int i = 0; i < m; i++) {
13         int x = a[i], y = b[i];
14         if ((col[x] ^ col[y]) == 1) {
15             if (col[x]) swap(x, y);
16             s[x][y] ^= 1, s[y][x] ^= 1;
17             col[x] = 1;
18             for (int j = 1; j <= n; j++) if (!col[j]) s[j][x] = 1, s[x][j] = 0;
19         }
20         else if ((col[x] & col[y]) == 1) s[x][y] ^= 1, s[y][x] ^= 1;
21     }
22     bool fg = false;
23     for (int i = 1; i <= n; i++) if (!col[i]) fg = true;
24     if (!fg) return 0;
25     for (int i = 0; i < m; i++) {
26         int x = a[i], y = b[i];
27         s[x][y] ^= 1, s[y][x] ^= 1;
28     }
29     for (int i = 1; i <= n; puts(""), i++) for (int j = i + 1; j <= n; j++) printf("%d ", s[i][j]);
30     return 1;
31 }
32
33 int main() {
34     int T;
35     scanf("%d", &T);
36     while (T--) {
37         scanf("%d %d", &n, &m);
38         for (int i = 0; i < m; i++) scanf("%d %d", &a[i], &b[i]);
39         for (int i = 1; i <= n; i++) if (check(i)) break;
40     }
41     return 0;
42 }
43 }
```

SOLUTION:

Status: 1 of 1 Correct Answer			Submission ID: 1000000000	
Score	Time	Memory		
100	0.00s	0.304		
Test Case	Case #	Result		
1	1	AC (0.000000s)		
1	1	AC (0.000000s)		
1	2	AC (0.000000s)		
Subtask Score: 30.00%			Result: AC	
2	1	AC (0.000000s)		
2	4	AC (0.000000s)		
2	5	AC (0.000000s)		
2	6	AC (0.000000s)		
Subtask Score: 40.00%			Result: AC	
3	1	AC (0.000000s)		
3	2	AC (0.000000s)		
3	3	AC (0.000000s)		
3	10	AC (0.000000s)		
3	11	AC (0.000000s)		
Subtask Score: 30.00%			Result: AC	
Total Score: 100.00%				

Question 9.

MANGO MARKET

```
Language: C++14

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6      ios::sync_with_stdio(false);
7      cin.tie(nullptr);
8      int n, m;
9      cin >> n >> m;
10     long long sum = 0;
11     for (int i = 1; i <= n; i++) {
12         long long x;
13         cin >> x;
14         sum += x;
15     }
16     long long edges = (long long)m, unused = ((long long)n * (n - 1)) / 2LL - edges;
17     for (int i = 0; i < m; i++) {
18         int u, v;
19         cin >> u >> v;
20     }
21     int b=edges-unused;
22     int q;
23     cin >> q;
24     for (int i = 0; i < q; i++) {
25         char x;
26         cin >> x;
27         if (x == '?') {
28             cout << sum + edges-unused << '\n';
29             continue;
30         }
31         int u, v;
32         cin >> u >> v;
33         if (x == '+') {
34             edges++;unused--;
35         }
36         else if (x == '-') {
37             edges--;
38             unused++;
39         }
40     }
41 }
42 return 0;
43
44 }
```

SOLUTION:

Status: ✓ Correct Answer		Submission ID: 85054358
Time: 0.04s	Memory: 5.4M	

Question 10.

Question 8. ONE MORE WEIRD GAME

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // your code goes here
6      int t,i;
7      cin>>t;
8      for(i=0;i<t;++i)
9      {
10         int n,m;
11         cin>>n>>m;
12
13         cout<<(n-1)+(m-1)+(2)*(n-1)*(m-1)<<endl;
14     }
15     return 0;
16 }
```

SOLUTION:

Status: ✓ Correct Answer			Submission ID: 85054581
Score: 100	Time: 0.00s	Memory: 5.3M	
Sub-Task	Task #	Result (time)	
1	1	AC (0.003705)	
Subtask Score: 30.00%		Result - AC	
2	2	AC (0.003869)	
Subtask Score: 70.00%		Result - AC	
Total Score = 100.00%			