

WORKSHEET 7

Student Name: Vivek Kumar

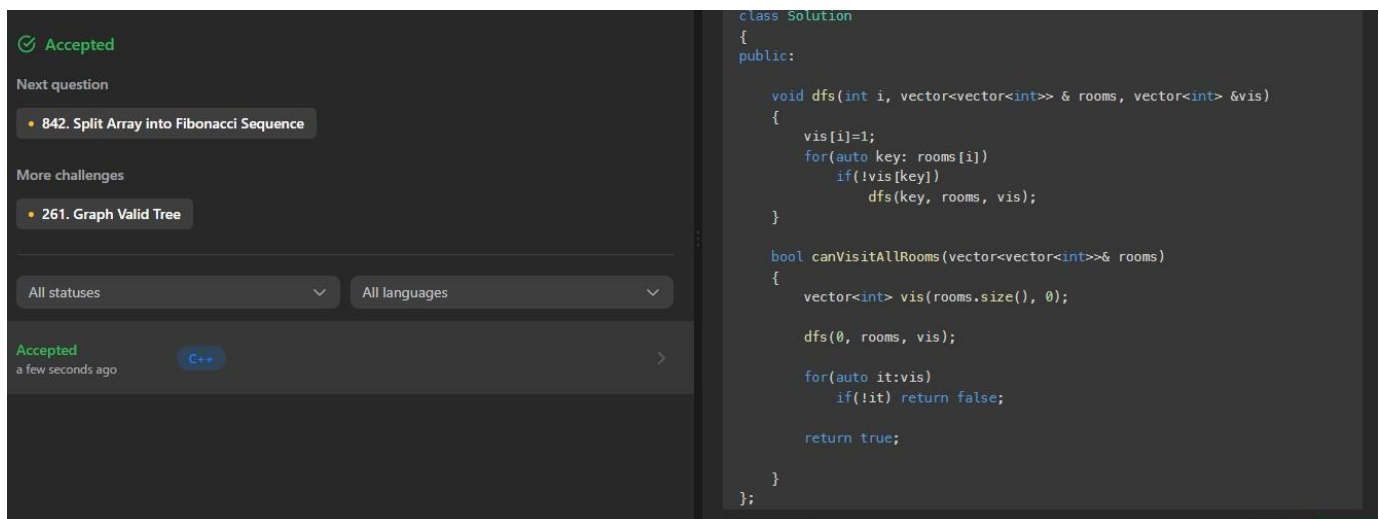
UID: 21BCS8129

DOMAIN CAMP: 16-01-2023 to 28-01-2023

Section/Group: DWWC-77

Subject Name: IT Skills (DSA)

Question 1. KEYS AND ROOMS



The screenshot displays a coding interface with a dark theme. On the left, a sidebar shows a list of challenges, including '842. Split Array into Fibonacci Sequence' and '261. Graph Valid Tree'. Below this, there are filters for 'All statuses' and 'All languages'. A green checkmark and the word 'Accepted' are visible, along with the text 'a few seconds ago' and a 'C++' button. The main area on the right shows the C++ code for the solution:

```
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:

1

Time:

0.02s

Memory:

5.4M

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:

Subtask 1 (100 points)			
Case	Time	Memory	Score
1	0.000	0.000	100
2	0.000	0.000	100
3	0.000	0.000	100
4	0.000	0.000	100
5	0.000	0.000	100
Subtask Score: 100%			
6	0.000	0.000	100
7	0.000	0.000	100
8	0.000	0.000	100
9	0.000	0.000	100
10	0.000	0.000	100
Subtask Score: 100%			
11	0.000	0.000	100
12	0.000	0.000	100
13	0.000	0.000	100
14	0.000	0.000	100
15	0.000	0.000	100
16	0.000	0.000	100
17	0.000	0.000	100
18	0.000	0.000	100
19	0.000	0.000	100
20	0.000	0.000	100
Subtask Score: 100%			
21	0.000	0.000	100
22	0.000	0.000	100
23	0.000	0.000	100
24	0.000	0.000	100
25	0.000	0.000	100
26	0.000	0.000	100
27	0.000	0.000	100
28	0.000	0.000	100
29	0.000	0.000	100
30	0.000	0.000	100
Subtask Score: 100%			
31	0.000	0.000	100
32	0.000	0.000	100
33	0.000	0.000	100
34	0.000	0.000	100
35	0.000	0.000	100
36	0.000	0.000	100
37	0.000	0.000	100
38	0.000	0.000	100
39	0.000	0.000	100
40	0.000	0.000	100
Subtask Score: 100%			
41	0.000	0.000	100
42	0.000	0.000	100
43	0.000	0.000	100
44	0.000	0.000	100
45	0.000	0.000	100
46	0.000	0.000	100
47	0.000	0.000	100
48	0.000	0.000	100
49	0.000	0.000	100
50	0.000	0.000	100
Subtask Score: 100%			
51	0.000	0.000	100
52	0.000	0.000	100
53	0.000	0.000	100
54	0.000	0.000	100
55	0.000	0.000	100
56	0.000	0.000	100
57	0.000	0.000	100
58	0.000	0.000	100
59	0.000	0.000	100
60	0.000	0.000	100
Subtask Score: 100%			
61	0.000	0.000	100
62	0.000	0.000	100
63	0.000	0.000	100
64	0.000	0.000	100
65	0.000	0.000	100
66	0.000	0.000	100
67	0.000	0.000	100
68	0.000	0.000	100
69	0.000	0.000	100
70	0.000	0.000	100
Subtask Score: 100%			
71	0.000	0.000	100
72	0.000	0.000	100
73	0.000	0.000	100
74	0.000	0.000	100
75	0.000	0.000	100
76	0.000	0.000	100
77	0.000	0.000	100
78	0.000	0.000	100
79	0.000	0.000	100
80	0.000	0.000	100
Subtask Score: 100%			
81	0.000	0.000	100
82	0.000	0.000	100
83	0.000	0.000	100
84	0.000	0.000	100
85	0.000	0.000	100
86	0.000	0.000	100
87	0.000	0.000	100
88	0.000	0.000	100
89	0.000	0.000	100
90	0.000	0.000	100
Subtask Score: 100%			
91	0.000	0.000	100
92	0.000	0.000	100
93	0.000	0.000	100
94	0.000	0.000	100
95	0.000	0.000	100
96	0.000	0.000	100
97	0.000	0.000	100
98	0.000	0.000	100
99	0.000	0.000	100
100	0.000	0.000	100
Subtask Score: 100%			
101	0.000	0.000	100
102	0.000	0.000	100
103	0.000	0.000	100
104	0.000	0.000	100
105	0.000	0.000	100
106	0.000	0.000	100
107	0.000	0.000	100
108	0.000	0.000	100
109	0.000	0.000	100
110	0.000	0.000	100
Subtask Score: 100%			
111	0.000	0.000	100
112	0.000	0.000	100
113	0.000	0.000	100
114	0.000	0.000	100
115	0.000	0.000	100
116	0.000	0.000	100
117	0.000	0.000	100
118	0.000	0.000	100
119	0.000	0.000	100
120	0.000	0.000	100
Subtask Score: 100%			
121	0.000	0.000	100
122	0.000	0.000	100
123	0.000	0.000	100
124	0.000	0.000	100
125	0.000	0.000	100
126	0.000	0.000	100
127	0.000	0.000	100
128	0.000	0.000	100
129	0.000	0.000	100
130	0.000	0.000	100
Subtask Score: 100%			
131	0.000	0.000	100
132	0.000	0.000	100
133	0.000	0.000	100
134	0.000	0.000	100
135	0.000	0.000	100
136	0.000	0.000	100
137	0.000	0.000	100
138	0.000	0.000	100
139	0.000	0.000	100
140	0.000	0.000	100
Subtask Score: 100%			
141	0.000	0.000	100
142	0.000	0.000	100
143	0.000	0.000	100
144	0.000	0.000	100
145	0.000	0.000	100
146	0.000	0.000	100
147	0.000	0.000	100
148	0.000	0.000	100
149	0.000	0.000	100
150	0.000	0.000	100
Subtask Score: 100%			
151	0.000	0.000	100
152	0.000	0.000	100
153	0.000	0.000	100
154	0.000	0.000	100
155	0.000	0.000	100
156	0.000	0.000	100
157	0.000	0.000	100
158	0.000	0.000	100
159	0.000	0.000	100
160	0.000	0.000	100
Subtask Score: 100%			
161	0.000	0.000	100
162	0.000	0.000	100
163	0.000	0.000	100
164	0.000	0.000	100
165	0.000	0.000	100
166	0.000	0.000	100
167	0.000	0.000	100
168	0.000	0.000	100
169	0.000	0.000	100
170	0.000	0.000	100
Subtask Score: 100%			
171	0.000	0.000	100
172	0.000	0.000	100
173	0.000	0.000	100
174	0.000	0.000	100
175	0.000	0.000	100
176	0.000	0.000	100
177	0.000	0.000	100
178	0.000	0.000	100
179	0.000	0.000	100
180	0.000	0.000	100
Subtask Score: 100%			
181	0.000	0.000	100
182	0.000	0.000	100
183	0.000	0.000	100
184	0.000	0.000	100
185	0.000	0.000	100
186	0.000	0.000	100
187	0.000	0.000	100
188	0.000	0.000	100
189	0.000	0.000	100
190	0.000	0.000	100
Subtask Score: 100%			
191	0.000	0.000	100
192	0.000	0.000	100
193	0.000	0.000	100
194	0.000	0.000	100
195	0.000	0.000	100
196	0.000	0.000	100
197	0.000	0.000	100
198	0.000	0.000	100
199	0.000	0.000	100
200	0.000	0.000	100
Subtask Score: 100%			
201	0.000	0.000	100
202	0.000	0.000	100
203	0.000	0.000	100
204	0.000	0.000	100
205	0.000	0.000	100
206	0.000	0.000	100
207	0.000	0.000	100
208	0.000	0.000	100
209	0.000	0.000	100
210	0.000	0.000	100
Subtask Score: 100%			
211	0.000	0.000	100
212	0.000	0.000	100
213	0.000	0.000	100
214	0.000	0.000	100
215	0.000	0.000	100
216	0.000	0.000	100
217	0.000	0.000	100
218	0.000	0.000	100
219	0.000	0.000	100
220	0.000	0.000	100
Subtask Score: 100%			
221	0.000	0.000	100
222	0.000	0.000	100
223	0.000	0.000	100
224	0.000	0.000	100
225	0.000	0.000	100
226	0.000	0.000	100
227	0.000	0.000	100
228	0.000	0.000	100
229	0.000	0.000	100
230	0.000	0.000	100
Subtask Score: 100%			
231	0.000	0.000	100
232	0.000	0.000	100
233	0.000	0.000	100
234	0.000	0.000	100
235	0.000	0.000	100
236	0.000	0.000	100
237	0.000	0.000	100
238	0.000	0.000	100
239	0.000	0.000	100
240	0.000	0.000	100
Subtask Score: 100%			
241	0.000	0.000	100
242	0.000	0.000	100
243	0.000	0.000	100
244	0.000	0.000	100
245	0.000	0.000	100
246	0.000	0.000	100
2			

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%		

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

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: ✓ Correct Answer			Submission ID: 1045411
Source	Time	Memory	
100	0.00s	1.3M	
Test Case	Test #	Result	Time
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	1	AC	0.000000s
2	1	AC	0.000000s
2	2	AC	0.000000s
Subtask Score: 40.00%			Result: AC
3	1	AC	0.000000s
3	1	AC	0.000000s
3	1	AC	0.000000s
3	2	AC	0.000000s
3	3	AC	0.000000s
Subtask Score: 30.00%			Result: AC
Total Score: 100.00%			

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
43 return 0;
44 }
```

SOLUTION:

Status: ✓ Correct Answer Submission ID: [85054358](#)

Time: 0.04s	Memory: 5.4M
----------------	-----------------

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%			