**BÁO CÁO PROJECT 1**

**WEEK 3**

Nội dung

[**Hoàn Thành 4/4** 2](#_Toc181109730)

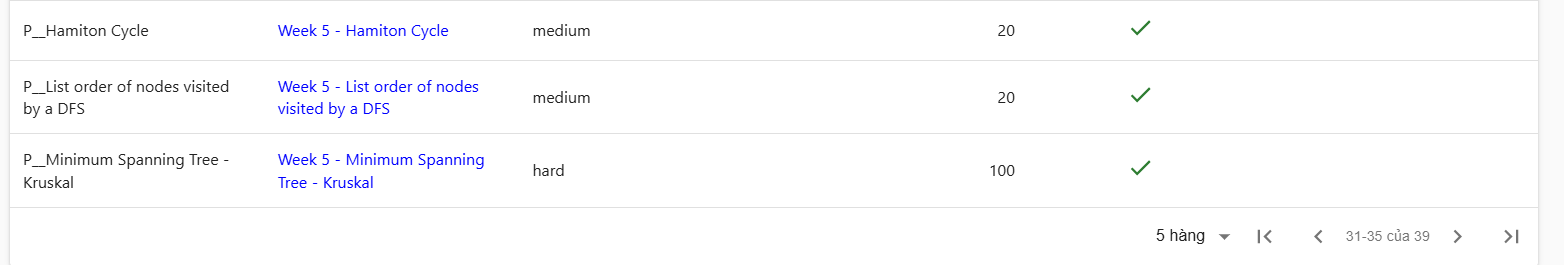
[**Problem: Week 5** HAM\_CYCLE 2](#_Toc181109731)

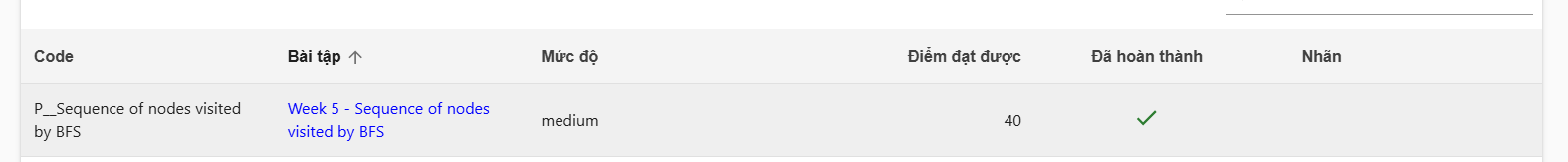
[**Problem: Week 5** DFS\_LIST\_ORDER\_NODES 5](#_Toc181109732)

[**Problem: Week 5 MST\_KRUSKAL** 7](#_Toc181109733)

[**Problem: Week 5** BFS\_LIST\_NODES\_LEX 10](#_Toc181109734)

# **Hoàn Thành 4/4**





# **Problem: Week 5** [HAM\_CYCLE](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/HAM_CYCLE)

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define maxn 300005

#define int long long

#define X first

#define Y second

using namespace std;

const int MAX = 20;

int adj[MAX][MAX];

bool visited\_arr[MAX];

int n, m;

bool hamiltonianCycleUtil(int current, int count)

{

if (count == n)

{

if (adj[current][0] == 1)

{

return true;

}

return false;

}

for(int next = 0; next < n; next++)

{

if(adj[current][next] && !visited\_arr[next])

{

visited\_arr[next] = true;

if(hamiltonianCycleUtil(next, count + 1))

{

return true;

}

visited\_arr[next] = false;

}

}

return false;

}

bool isHamiltonian()

{

memset(visited\_arr, false, sizeof(visited\_arr));

visited\_arr[0] = true;

return hamiltonianCycleUtil(0, 1);

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

int T;

cin >> T;

while(T--)

{

cin >> n >> m;

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

adj[i][j] = 0;

}

}

for(int i = 0; i < m; i++)

{

int u, v;

cin >> u >> v;

u--;

v--;

adj[u][v] = 1;

adj[v][u] = 1;

}

bool ans = isHamiltonian();

cout << ans <<"\n";

}

}

# **Problem: Week 5** [DFS\_LIST\_ORDER\_NODES](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/DFS_LIST_ORDER_NODES)

A screenshot of a phone

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define maxn 300005

#define int long long

using namespace std;

int n, m, tr[maxn];

vector<int> a[maxn];

void DFS(int u)

{

for(auto v : a[u])

{

if(tr[v] == 0)

{

tr[v] = 1;

cout << v <<" ";

DFS(v);

}

}

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

cin >> n >> m;

for(int i = 1; i <= m; i++)

{

int u, v;

cin >> u >> v;

a[u].push\_back(v);

a[v].push\_back(u);

}

tr[1] = 1;

cout << 1 <<" ";

DFS(1);

}

# **Problem: Week 5** [**MST\_KRUSKAL**](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/MST_KRUSKAL)

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define maxn 300005

#define int long long

#define X first

#define Y second

using namespace std;

typedef pair<int, int> ii;

typedef pair<int, ii> iii;

int par[230997];

int anc(int p)

{

if (par[p] == p)

return p;

else

return par[p] = anc(par[p]);

}

void join(int p, int q)

{

par[anc(p)] = anc(q);

}

vector<iii> edge;

int n, m;

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

cin >> n >> m;

for (int i = 1; i <= n; i++)

par[i] = i;

while (m--)

{

int p, q, w;

cin >> p >> q >> w;

edge.push\_back(iii(w, ii(p, q)));

}

sort(edge.begin(), edge.end());

int r = 0;

for (auto it = edge.begin(); it != edge.end(); it++)

{

if (anc(it->Y.X) != anc(it->Y.Y))

{

join(it->Y.X, it->Y.Y);

r += it->X;

}

}

cout << r << "\n";

return 0;

}

# **Problem: Week 5** [BFS\_LIST\_NODES\_LEX](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/BFS_LIST_NODES_LEX)

A screenshot of a phone

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define maxn 300005

#define int long long

using namespace std;

int n, m, tr[maxn];

vector<int> a[maxn];

void BFS(int u)

{

queue<int> qu;

qu.push(u);

tr[u] = 1;

vector<int> result;

result.push\_back(u);

while (!qu.empty())

{

int u1 = qu.front();

qu.pop();

sort(a[u1].begin(), a[u1].end());

for (auto v : a[u1])

{

if (tr[v] == 0)

{

tr[v] = 1;

qu.push(v);

result.push\_back(v);

}

}

}

for (int node : result)

{

cout << node << " ";

}

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

cin >> n >> m;

for (int i = 1; i <= m; i++)

{

int u, v;

cin >> u >> v;

a[u].push\_back(v);

a[v].push\_back(u);

}

//BFS(1);

for(int i = 1; i <= n; i++)

{

if(tr[i] == 0)

{

BFS(i);

}

}

return 0;

}