LECTURE 3

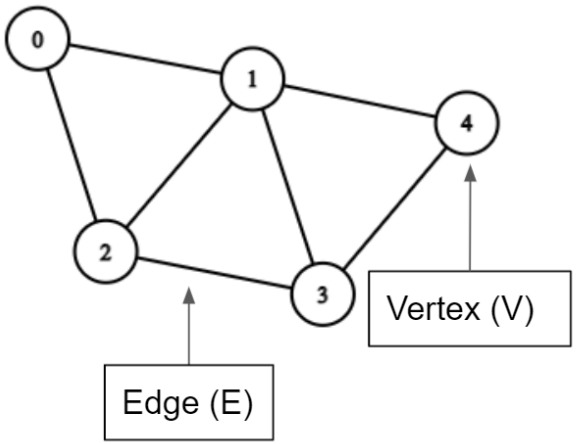
BASIC GRAPH REDUCTION

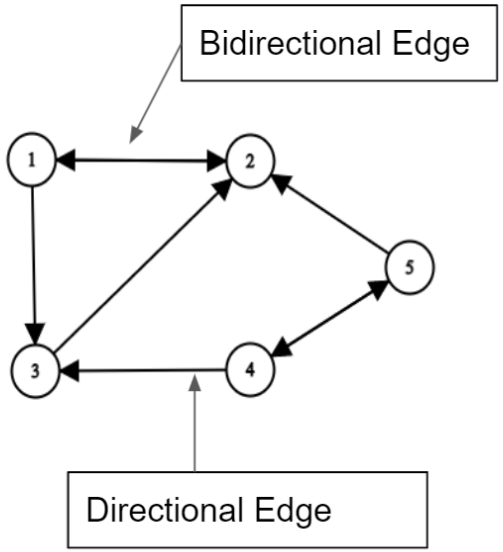
Benjamin Li & Po Hao Chen

September 24, 2021

Definition

Our View of A Graph

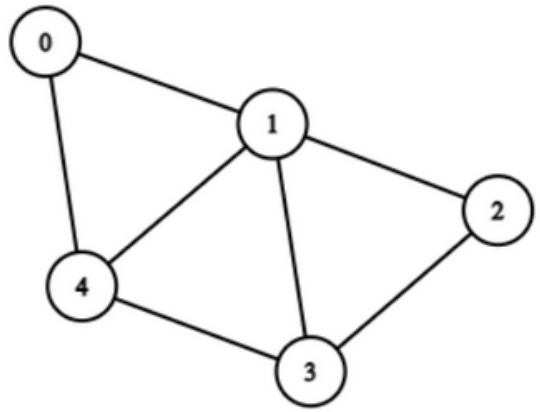




Code Representation

Can represent graph in two ways.

1. Adjacency Matrix



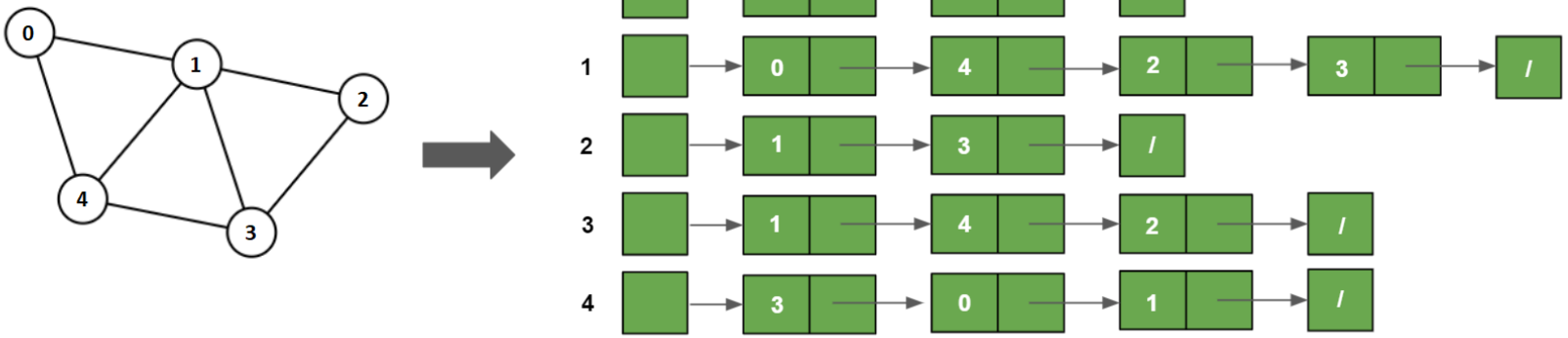
Node	0	1	2	3	4
0	0	1	1	0	0
1	1	0	1	1	1
2	1	1	0	1	0
3	0	1	1	0	1
4	0	1	0	1	0

Adding a node and graph traversal are $O(V^2)$.

$M_{u,v} = 1$ iff u has a directed edge to v

Code Representation

2. Adjacency List (List of LinkedLists)



Complexity: `O(V + E)`

Adjacency List in C++

```cpp
vector<int> adj[10001];

int main(){
    int n,m;
    cin >> n >> m; // read |V| and |E|

    for(int i = 0 ; i < m ; i++){ // Read m edges
        int a,b;
        cin >> a >> b; //Read connectivity between node a & node b
        adj[a].push_back(b);
        adj[b].push_back(a); // If undirected must add both ways
    }
}
```

DFS & BFS

Depth-First Search (DFS)

- Explore as deep as possible, backtrack once branch is fully explored.
- Implement recursively or with a stack.

```
int vis[1001];
void dfs(int v){
    vis[v] = 1; //mark it upon visit
    for(auto &i : adj[v]){
        if(!vis[i]) //DFS on all unvisited neighbors
            dfs(i);
    }
}
```

Breadth-First Search (BFS)

- Explore as wide as possible, look through all neighbors before moving on.
- Implement using a queue.

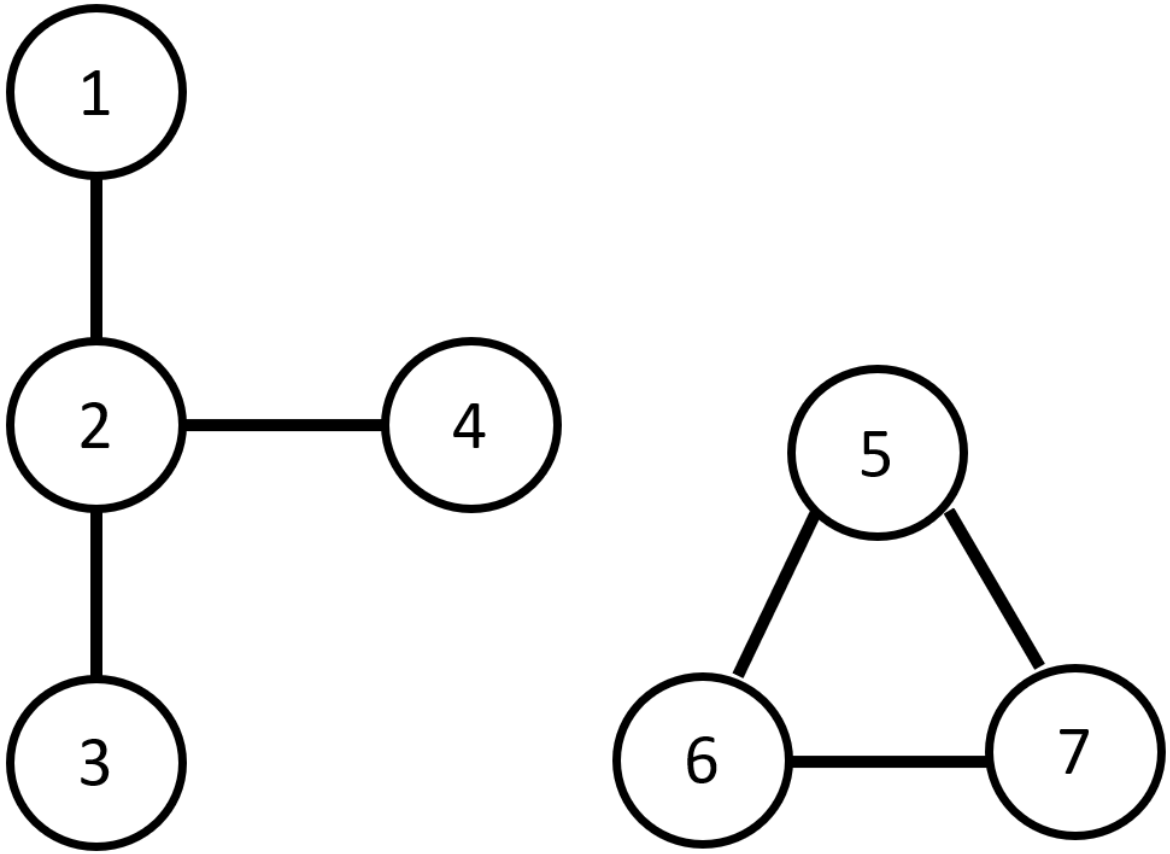
```
int source = 0; //start at node 0

queue<int> q;
q.push(source);
vis[source] = 1; //mark source as visited

while(!q.empty()){
    int v = q.front();
    q.pop();
    for(auto &u : adj[v]){
        if(!vis[u]){
            vis[u] = 1;
            q.push(u);
            dist[u] = dist[v]++;
        }
    }
}
```

BFS is particularly good for finding the shortest path in an unweighted graph.

Connected Components



Graphs can have multiple components that do not touch.

Graph Reduction

Molecule Interactions

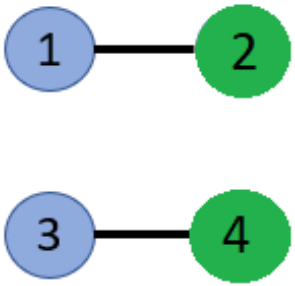
Problem: A pharmaceutical company has two sets of molecules and wants to test a hypothesis: whether or not each molecule only interacts with molecules from the other group. Output "Yes" if the hypothesis holds (there are no intra-interactions) or "No" otherwise.

Input: A single row containing n, the of molecules, and m, the number of interactions. m rows will follow, each containing two numbers representing an interaction between molecules.

Example

Example 1

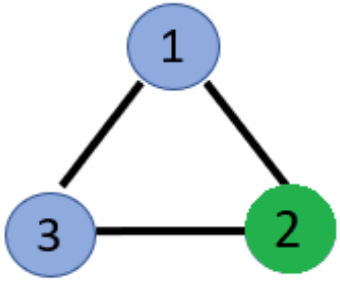
Input:
4 2
1 2
3 4



Output:
Yes

Example 2

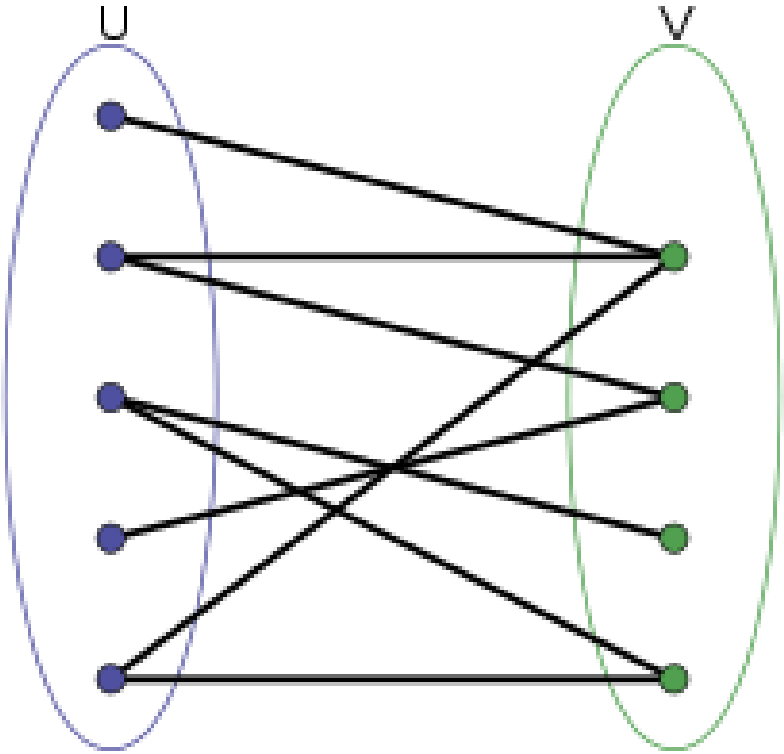
Input:
3 3
1 2
2 3
3 1



Output:
No

Observation

We can separate the two groups, circle them, and map each interaction. If we find that no two molecules of the same group have an interaction, we know the answer! This is a test of bipartiteness.



Theorem: A graph is bipartite iff it is two-colorable.

Prime Path

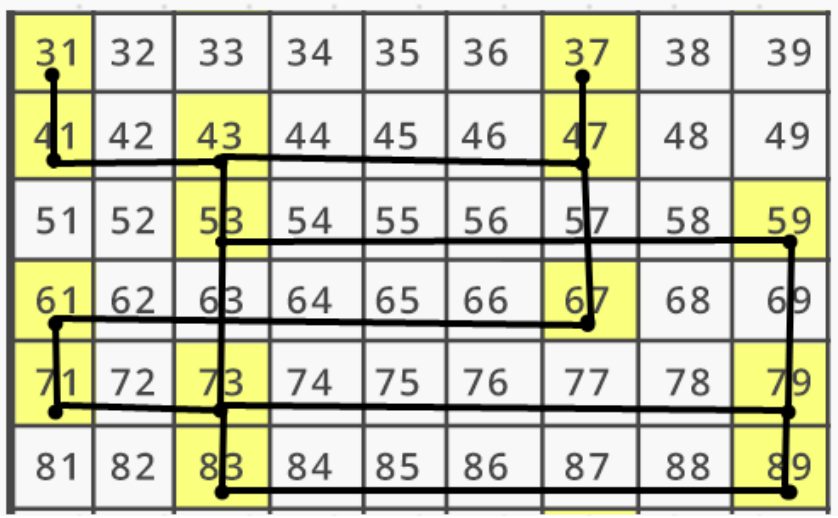
Given two 4-digit prime numbers x and y, what is the minimum amount of steps needed to change x into y? You can modify one digit each time to any digit of your choice, but each intermediate number also has to be prime.

Input: 1033 8179
Output: 6 steps
Explanation: $1033 \rightarrow 1733 \rightarrow 3733 \rightarrow 3739 \rightarrow 3779 \rightarrow 8779 \rightarrow 8179$

Prime Path

Note: This is an example with 2-digit numbers

- Need to do two things first:
  - Find all prime numbers
  - Identify all those that are 1 change away



Prime Path

Explore with BFS to find the shortest path