

I ILLINOIS

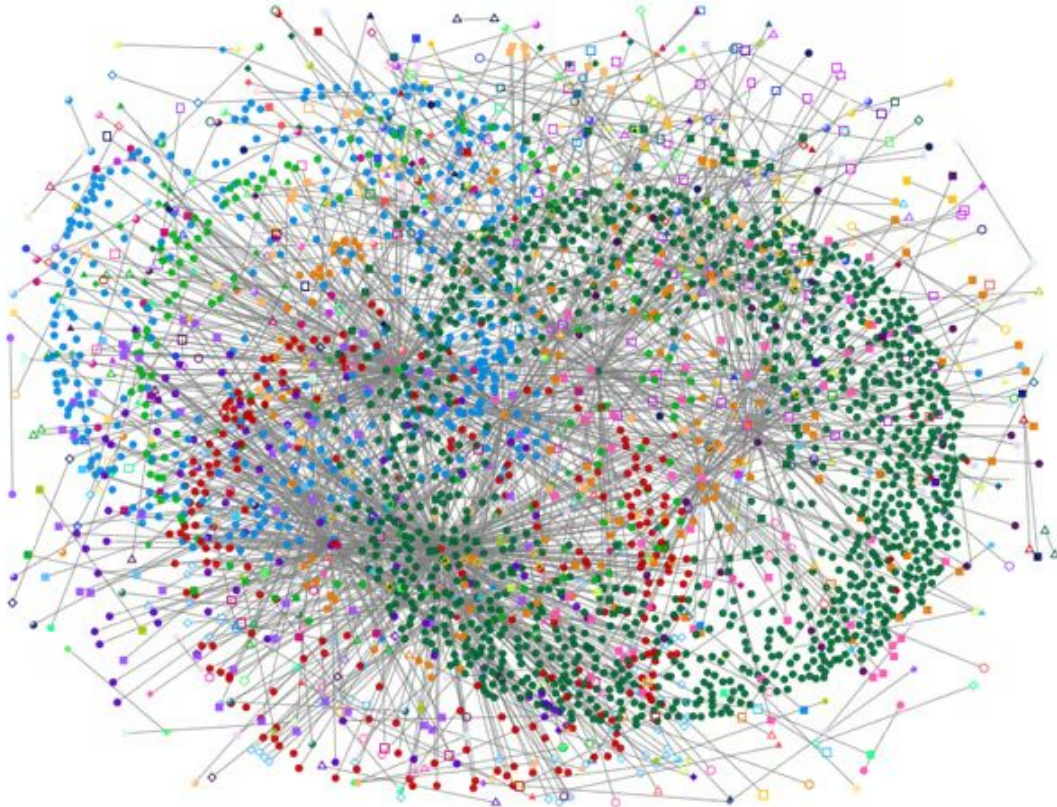
Graph Implementation

Learning Objectives

1. Understand the Edge List Implementation of a Graph



Strings -> Compact Integers

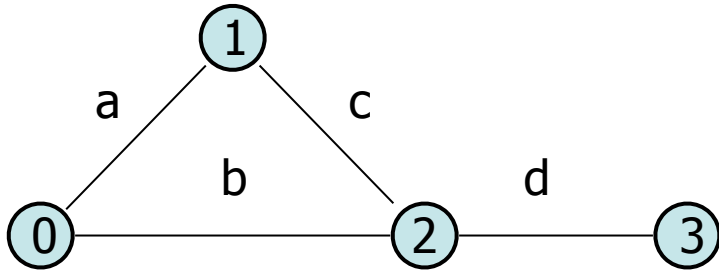


Similar to Disjoint Sets

- External Data Structure Used
- Especially for large graphs

Graph Implementation - Edge List

$G = (V, E)$



0
1
2
3

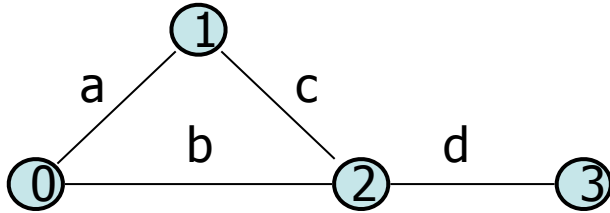
Vertices

0	1	a
0	2	b
1	2	c
2	3	d

Edges

Graph Implementation - Edge List

$G = (V, E)$



0	0	1	a
1	0	2	b
2	1	2	c
3	2	3	d

`insertVertex(K key);`

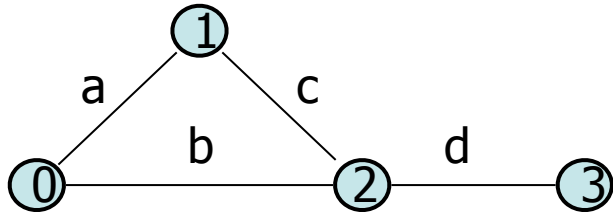
`insertEdge(Vertex v1, Vertex v2, K key);`

Vertices Edges



Graph Implementation - Edge List

$G = (V, E), |V| = n, |E| = m$



0	0	1	a
1	0	2	b
2	1	2	c
3	2	3	d

`removeVertex(Vertex v);`

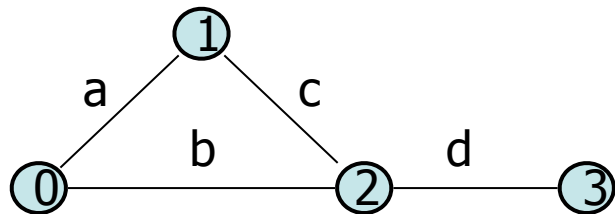
`removeEdge(Vertex v1, Vertex v2);`

Vertices Edges



Graph Implementation - Edge List

$G = (V, E), |V| = n, |E| = m$



0	0	1	a
1	0	2	b
2	1	2	c
3	2	3	d

`incidentEdges(Vertex v);`

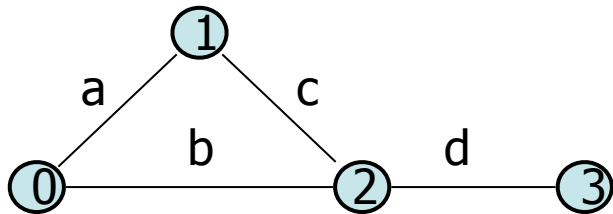
`areAdjacent(Vertex v1, Vertex v2);`

Vertices Edges



Graph Implementation - Edge List

$G = (V, E), |V| = n, |E| = m$



0
1
2
3

0	1	a
0	2	b
1	2	c
2	3	d

insertVertex(K key); - $O(1)$

insertEdge(Vertex v1, Vertex v2, K key); - $O(1)$

removeVertex(Vertex v); - $O(m)$

removeEdge(Vertex v1, Vertex v2); - $O(m)$

incidentEdges(Vertex v); - $O(m)$

areAdjacent(Vertex v1, Vertex v2); - $O(m)$

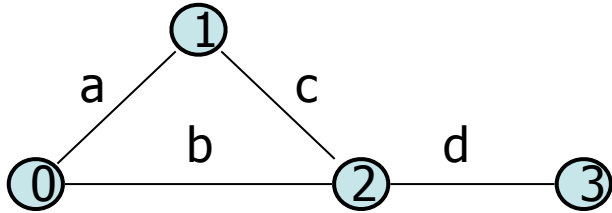
Vertices Edges



Graph Implementation - Edge List

$$G = (V, E), |V| = n, |E| = m$$

Great for Storing and Loading Graphs!



0	0	1	a
1	0	2	b
2	1	2	c
3	2	3	d

Vertices Edges

