AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

22BIO211: Intelligence of Biological Systems - 2

GRAPH REPRESENTATIONS

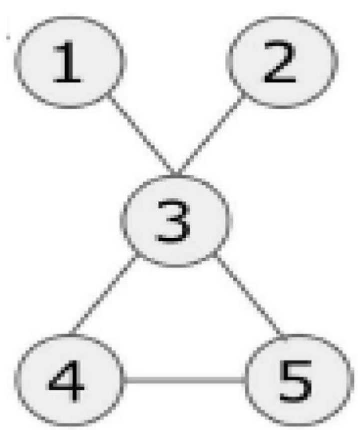
Dr. Manjusha Nair M
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Email : manjushanair@am.amrita.edu
Contact No: 9447745519

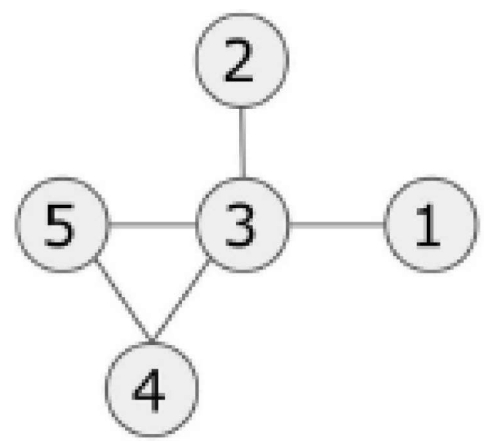
Different Graphical Representations for the Same Graph G = (V,E)

Given

Vertex set $\boldsymbol{V} = \{1, 2, 3, 4, 5\}$

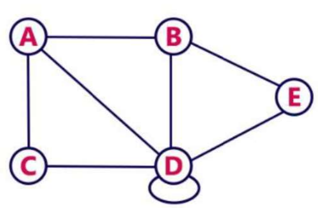
Edge set $\boldsymbol{E} = \{\{1, 3\}, \{2, 3\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$





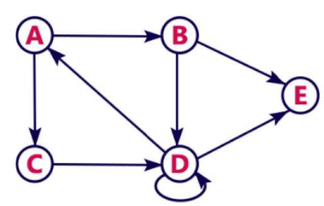
Representing Graphs

- Adjacency Matrix
  - Using an M X M matrix, where M is the number of vertices
  - A value 1 represents the presence of an edge, 0 otherwise.



$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 1 & 1 & 0 \\ B & 1 & 0 & 0 & 1 & 1 \\ C & 1 & 0 & 0 & 1 & 0 \\ D & 1 & 1 & 1 & 1 & 1 \\ E & 0 & 1 & 0 & 1 & 0 \end{array}$$

Undirected graph representation.



$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 1 & 0 & 0 \\ B & 0 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 0 \\ D & 1 & 0 & 0 & 1 & 1 \\ E & 0 & 0 & 0 & 0 & 0 \end{array}$$

Directed graph representation.

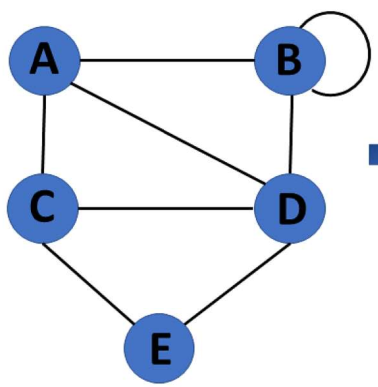
Representing Graphs

- **1. Adjacency Matrix**
- In a unweighted graph the element A[i][j] represents the Boolean value that determine whether a path exist from vertex i to vertex j
- In a weighted graph the element A[i][j] represents the cost of moving vertex i to vertex j

Advantages	**Disadvantages**
Space-efficient for dense graph representation.	Space Complexity of this Data Structure - $O(V^2)$.
The time complexity of getting an edge weight is $O(1)$.	Iterating through the edges takes $O(V^2)$ time.
Simplest Graph Representation.	

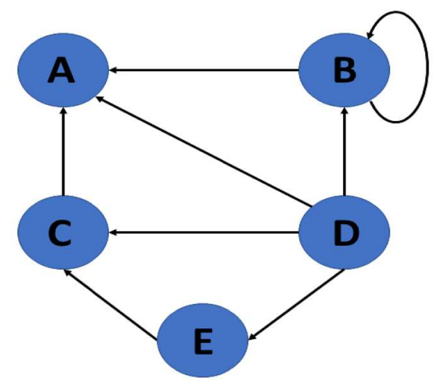
Representing Graphs - Examples

Undirected Graph Representation



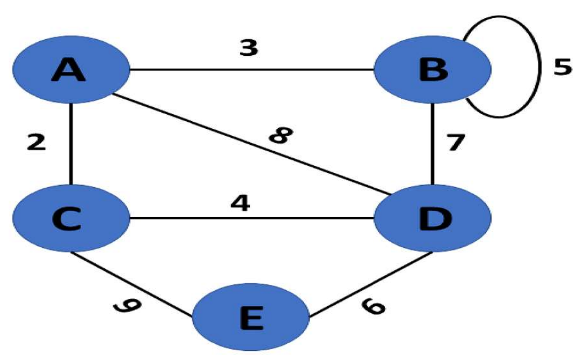
$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 1 & 1 & 0 \\ B & 1 & 1 & 0 & 1 & 0 \\ C & 1 & 0 & 0 & 1 & 1 \\ D & 1 & 1 & 1 & 0 & 1 \\ E & 0 & 0 & 1 & 1 & 0 \end{array}$$

Directed Graph Representation



$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 0 & 0 & 0 & 0 \\ B & 1 & 1 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 0 & 0 \\ D & 1 & 1 & 1 & 0 & 1 \\ E & 0 & 0 & 1 & 0 & 0 \end{array}$$

Weighted Undirected Graph Representation



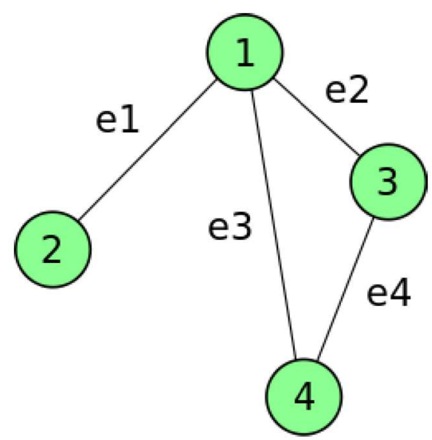
$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 3 & 2 & 8 & 0 \\ B & 3 & 5 & 0 & 7 & 0 \\ C & 2 & 0 & 0 & 4 & 9 \\ D & 8 & 7 & 4 & 0 & 6 \\ E & 0 & 0 & 9 & 6 & 0 \end{array}$$

Representing Graphs

- 2. **Incidence Matrix**
  - Using an M X N matrix, where M is the number of vertices and N is the number of edges
  - rows represent vertices and columns represents edges

Incidence matrix of the undirected graph

$$B_{ij} = \begin{cases} 1 & \text{if vertex } v_i \text{ is incident with edge } e_j, \\ 0 & \text{otherwise.} \end{cases}$$



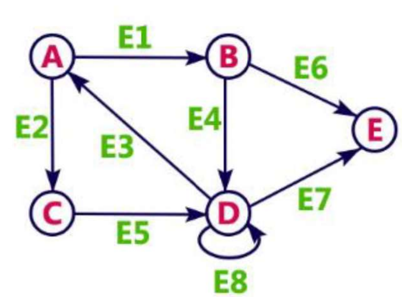
	e_1	e_2	e_3	e_4
1	1	1	1	0
2	1	0	0	0
3	0	1	0	1
4	0	0	1	1

$$= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Representing Graphs

Incidence matrix of a directed graph

- A value O represents no edge, 1 represents outgoing edge, -1 represents incoming edge.

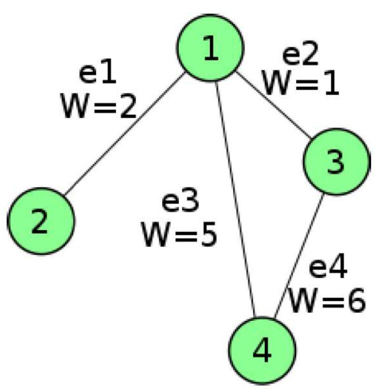


$$
\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array}
\begin{array}{c}
\begin{array}{cccccccc} E1 & E2 & E3 & E4 & E5 & E6 & E7 & E8 \end{array} \\
\begin{pmatrix}
1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & -1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 0
\end{pmatrix}
\end{array}
$$

Representing Graphs

Incidence matrix of weighted graph

- A weighted graph can be represented using the weight of the edge in place of a 1

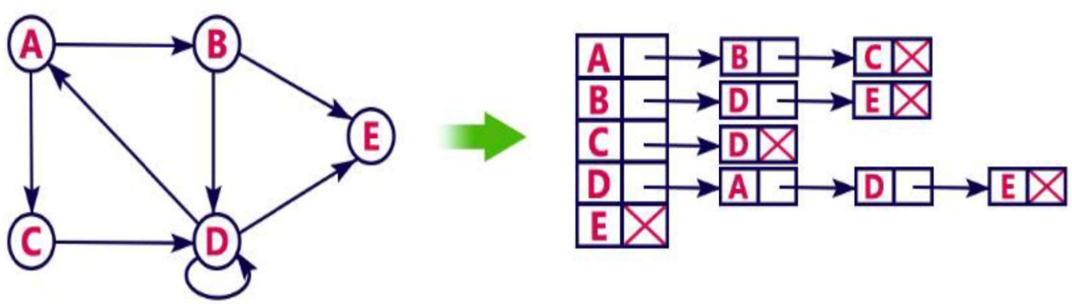


	e_1	e_2	e_3	e_4
1	2	1	5	0
2	2	0	0	0
3	0	1	0	6
4	0	0	5	6

$$= \begin{bmatrix} 2 & 1 & 5 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 5 & 6 \end{bmatrix}.$$

Representing Graphs

- 3. **Adjacency List**
  - Data Structure Linked List is used for representing graphs
  - Every vertex of a graph contains list of its adjacent vertices
  - For each node, we maintain a linked list of its neighbors.

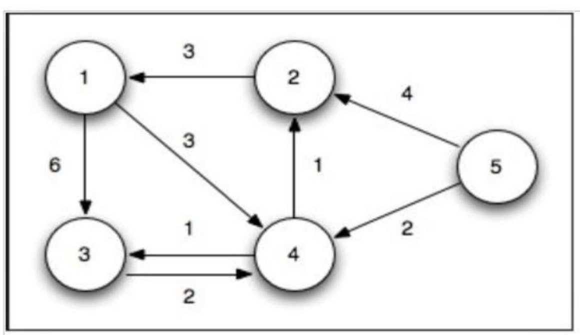


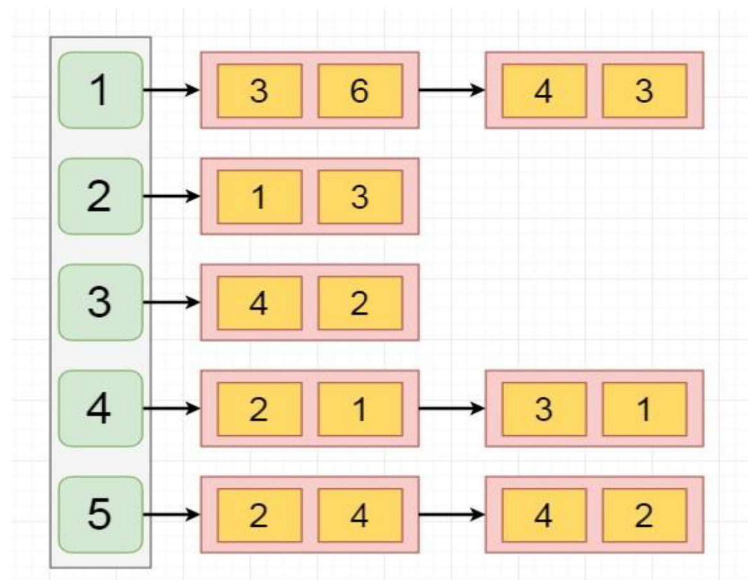
Advantages	Disadvantages
Space-efficient for sparse graphs.	Less space efficient for dense graphs.
Iterating over the edges is efficient.	Edge weight lookup is O(E). (worse case)
	Slightly more complex to represent.

Representing Graphs

Weighted directed Graph- Representation Using Linked-List

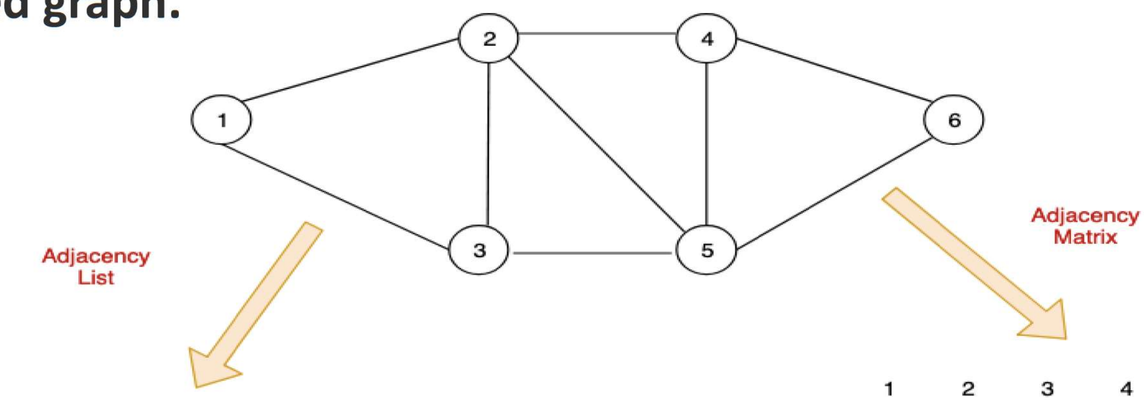
- In the adjacency list, each element in the list will have two data.
- The first one is the destination node, and the second one is the weight between these two nodes.
- The representation is like below.

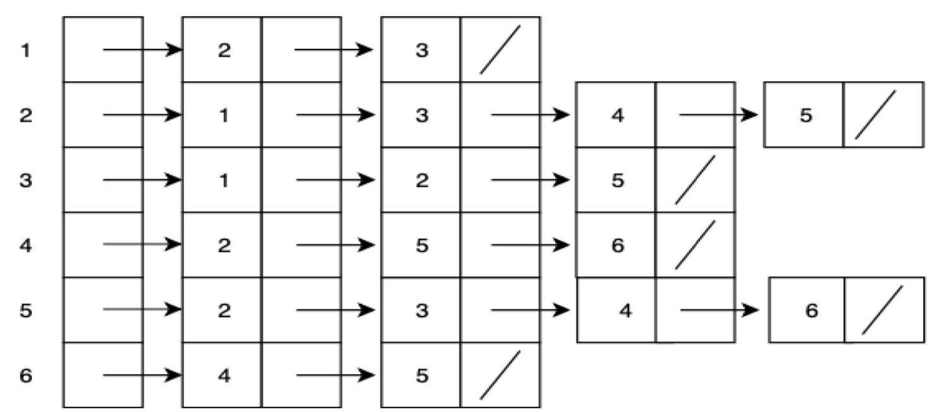




Representing Graphs

Example - Undirected graph.

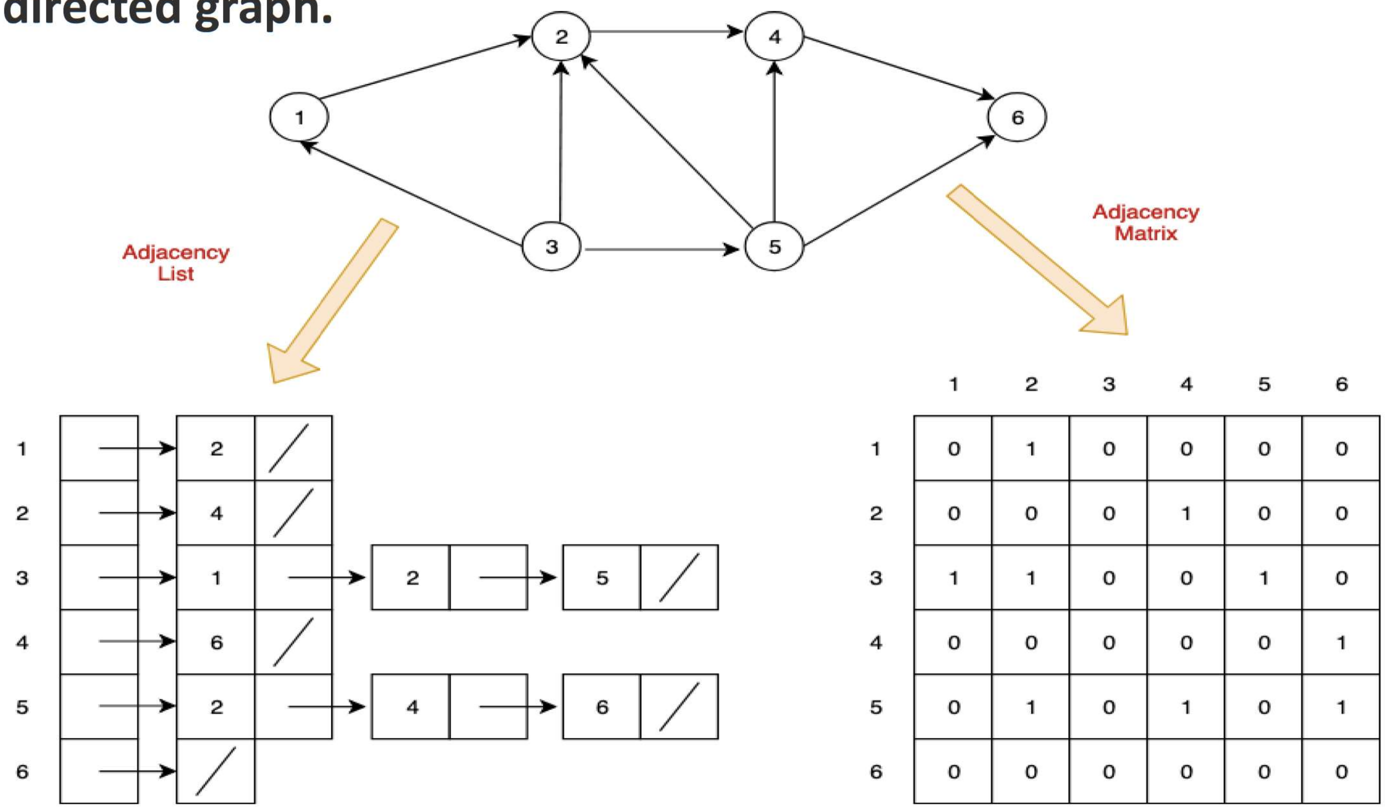




	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	1	1	0
3	1	1	0	0	1	0
4	0	1	0	0	1	1
5	0	1	1	1	0	1
6	0	0	0	1	1	0

Representing Graphs

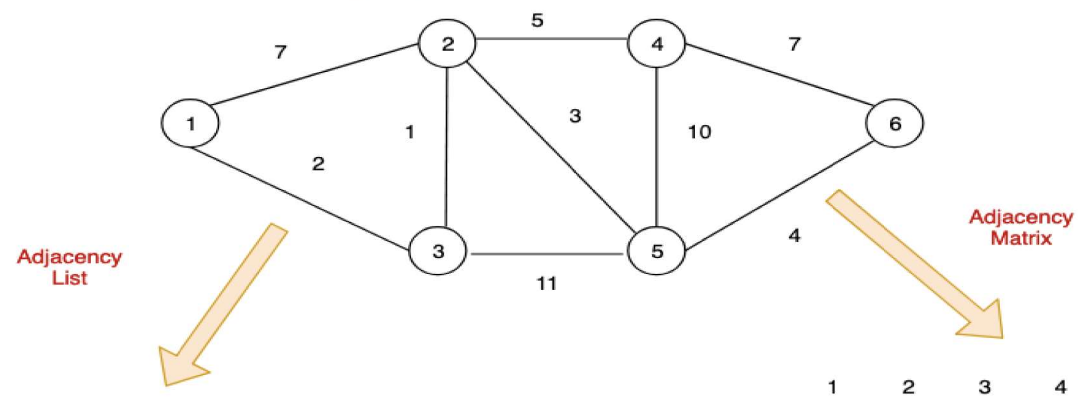
Example - directed graph.

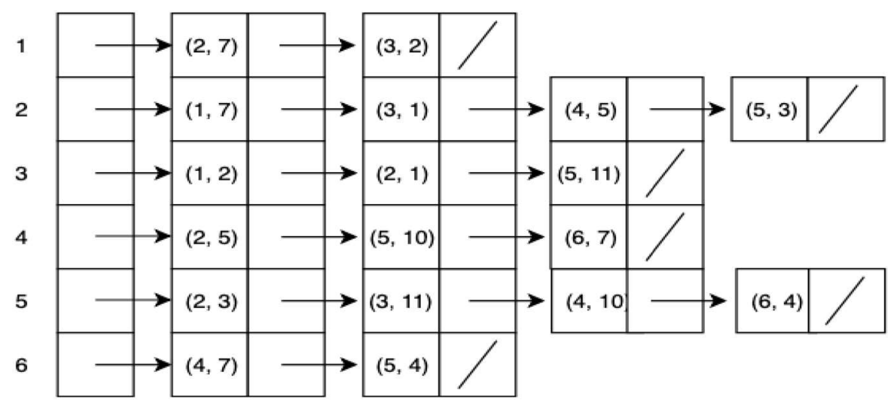


	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	0	1	0	0
3	1	1	0	0	1	0
4	0	0	0	0	0	1
5	0	1	0	1	0	1
6	0	0	0	0	0	0

Representing Graphs

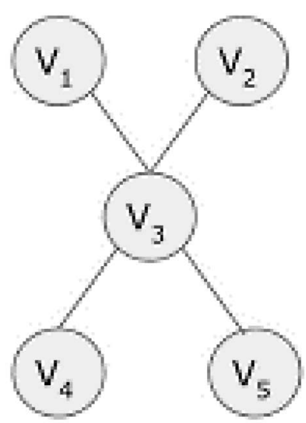
Example – Weighted undirected graph.





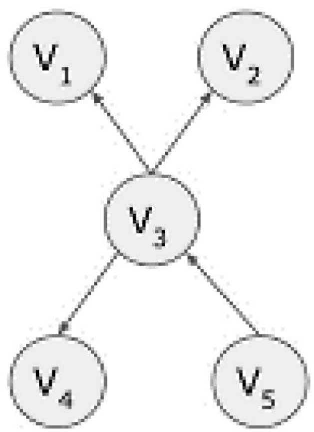
	1	2	3	4	5	6
1	0	7	2	0	0	0
2	7	0	1	5	3	0
3	2	1	0	0	11	0
4	0	5	0	0	10	7
5	0	3	11	10	0	4
6	0	0	0	7	4	0

Adjacency Matrix Representation : Undirected Graph



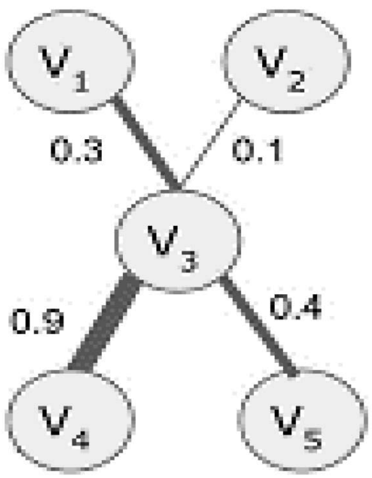
	V_1	V_2	V_3	V_4	V_5
V_1	0	0	1	0	0
V_2	0	0	1	0	0
V_3	1	1	0	1	1
V_4	0	0	1	0	0
V_5	0	0	1	0	0

Adjacency Matrix Representation : Directed Graph



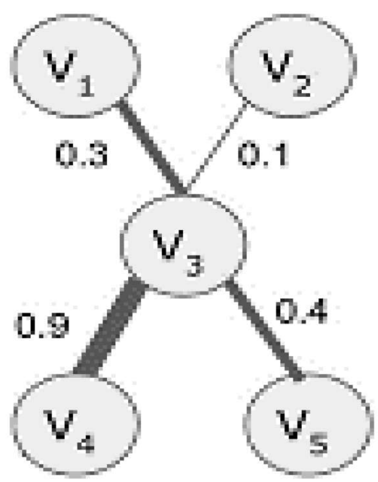
	V_1	V_2	V_3	V_4	V_5
V_1	0	0	0	0	0
V_2	0	0	0	0	0
V_3	1	1	0	1	0
V_4	0	0	0	0	0
V_5	0	0	1	0	0

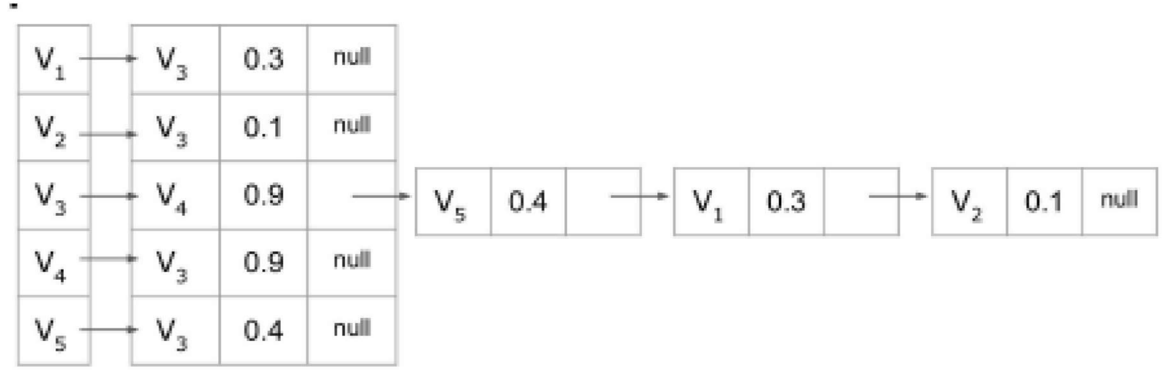
Adjacency Matrix Representation : Weighted Graph



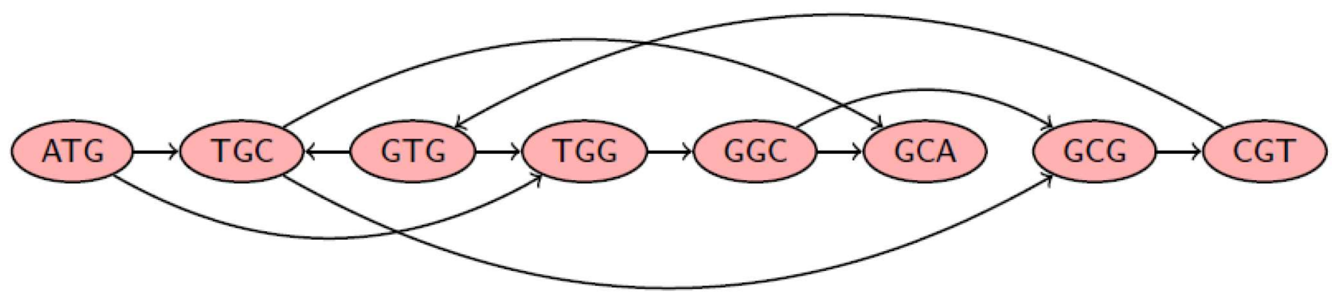
	V_1	V_2	V_3	V_4	V_5
V_1	0	0	0.3	0	0
V_2	0	0	0.1	0	0
V_3	0.3	0.1	0	0.9	0.4
V_4	0	0	0.9	0	0
V_5	0	0	0.4	0	0

Adjacency List Representation : Weighted Graph





Adjacency List Representation : Overlap Graph

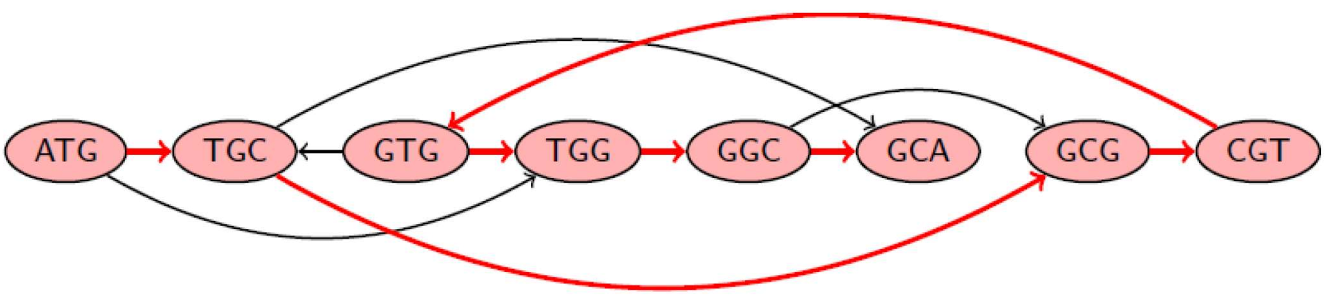


ATG $\rightarrow$ TGC, TGG
TGC $\rightarrow$ GCA, GCG
GTG $\rightarrow$ TGC, TGG
TGG $\rightarrow$ GGC
GGC $\rightarrow$ GCA, GCG
GCA $\rightarrow$
GCG $\rightarrow$ CGT
CGT $\rightarrow$ GTG

Hamiltonian Path in Overlap Graph

A Hamiltonian path in a graph is a path that visits every vertex exactly once.

A Hamiltonian path in an overlap graph lists the k-mers in an order they appear in some string

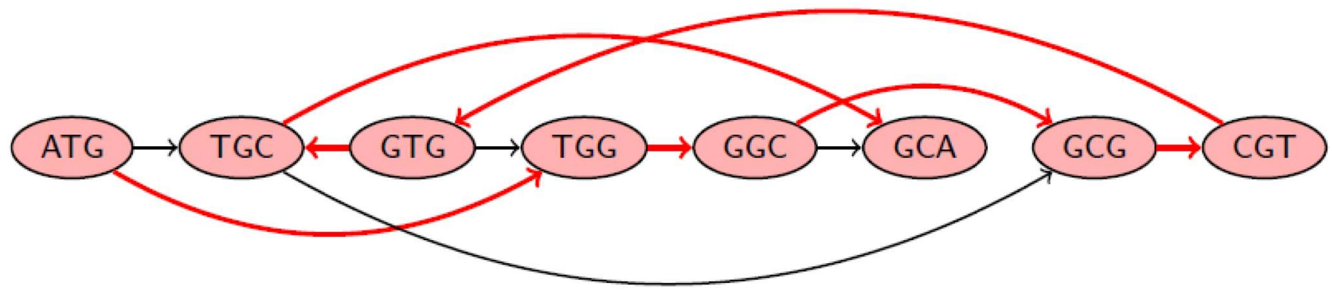


overlap graph

```
A T G
  T G C
    G C G
      C G T
        G T G
          T G G
            G G C
              G C A
A T G C G T G G C A
```

Another Hamiltonian Path

overlap graph




```
A T G
  T G G
    G G C
      G C G
        C G T
          G T G
            T G C
              G C A
A T G G C G T G C A
```

Summary

- Graph Representations
  - *Adjacency matrix*
  - *Incidence matrix*
  - *Adjacency list*
- Overlap Graphs