Discrete Structures

Fight for Survival

Dr. Mudassir Shabbir

August 27, 2026

Graph Representations

Graph Representations

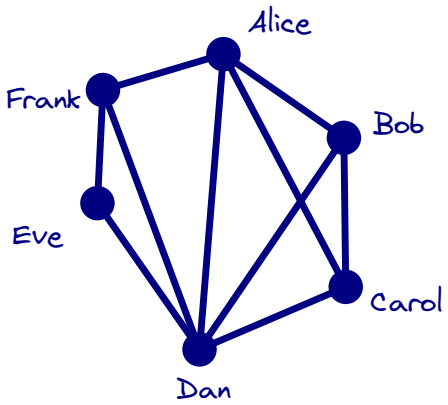
A representation is needed to:

- -store graphs in memory
- -share graphs information

Graph Representations

A representation is needed to:

- -store graphs in memory
- -share graphs information



We could just share this picture?

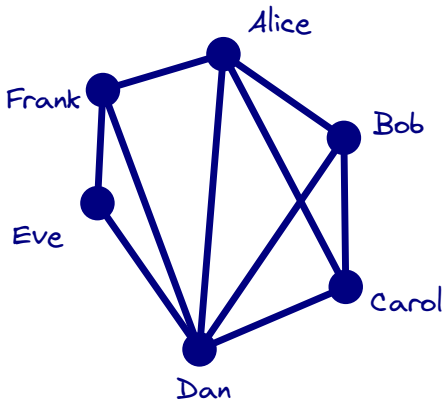
Graph Representations

A representation is needed to:

- store graphs in memory
- share graphs information

But it gets messy pretty fast!

We could just share this picture?



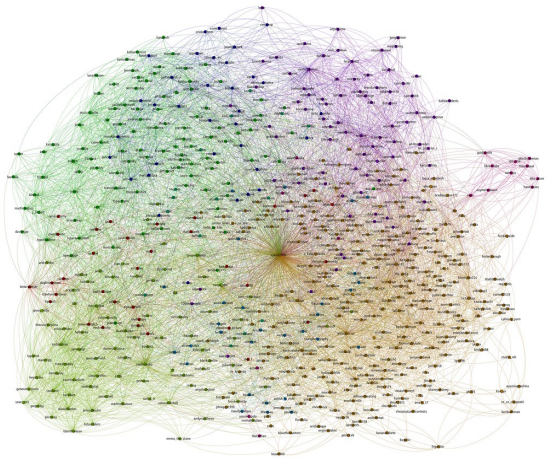
Graph Representations

A representation is needed to:

- store graphs in memory
- share graphs information

But it gets messy pretty fast!

We could just share this picture?



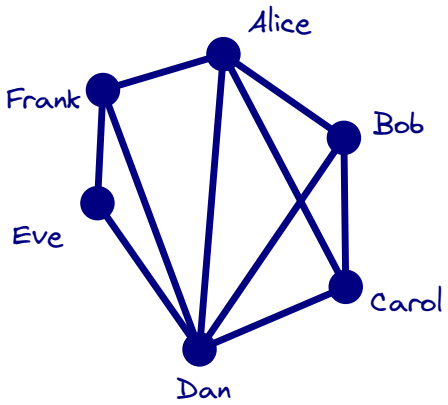
Graph Representations

A representation is needed to:

- store graphs in memory
- share graphs information

But it gets messy pretty fast!

We could just share this picture?



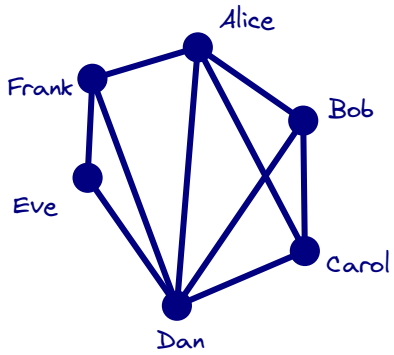
Three common ways to represent:

1. list of all edges
2. separate lists of edges for each vertex
3. yes/no for each possible edge

Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

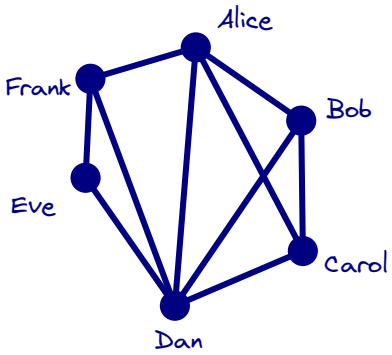
1. Edge List



Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

1. Edge List



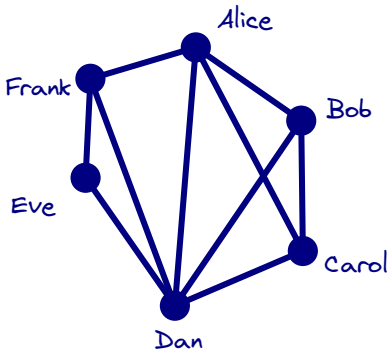
Alice i.f.w. Bob
Bob i.f.w. Carol
Carol i.f.w. Dan
Dan i.f.w. Eve
Eve i.f.w. Frank
Alice i.f.w. Frank
Alice i.f.w. Carol
Alice i.f.w. Dan
Bob i.f.w. Dan
Dan i.f.w. Frank

Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

1. Edge List

Easy to store but inefficient!



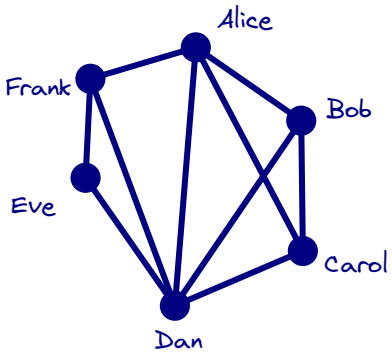
Alice i.f.w. Bob
Bob i.f.w. Carol
Carol i.f.w. Dan
Dan i.f.w. Eve
Eve i.f.w. Frank
Alice i.f.w. Frank
Alice i.f.w. Carol
Alice i.f.w. Dan
Bob i.f.w. Dan
Dan i.f.w. Frank

Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

1. Edge List

Easy to store but inefficient!



Alice i.f.w. Bob
Bob i.f.w. Carol
Carol i.f.w. Dan
Dan i.f.w. Eve
Eve i.f.w. Frank
Alice i.f.w. Frank
Alice i.f.w. Carol
Alice i.f.w. Dan
Bob i.f.w. Dan
Dan i.f.w. Frank

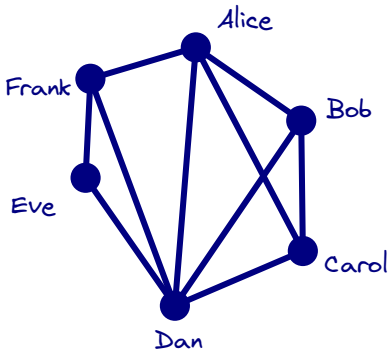
Need to go through the whole list to check wheher
Dan and Frank are friends?
OR
even to check if Eve has any friends?

Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

1. Edge List

Easy to store but inefficient!



Alice i.f.w. Bob
Bob i.f.w. Carol
Carol i.f.w. Dan
Dan i.f.w. Eve
Eve i.f.w. Frank
Alice i.f.w. Frank
Alice i.f.w. Carol
Alice i.f.w. Dan
Bob i.f.w. Dan
Dan i.f.w. Frank

Need to go through the whole list to check wheher Dan and Frank are friends?
OR
even to check if Eve has any friends?

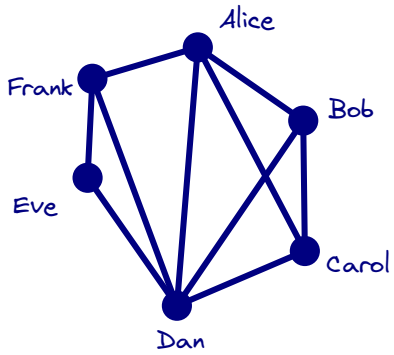
Commonly used in Machine Learning algorithms that deal with graphs with trillions of edges.

Graph Representations

A representation is needed to:

- -store graphs in memory
- -share graphs information

2. Adjacency List



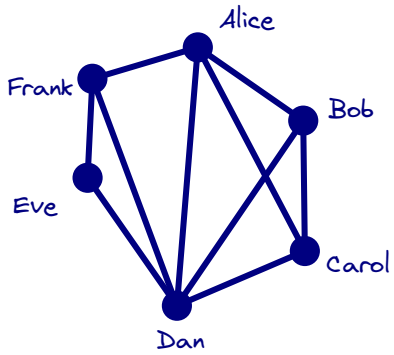
Graph Representations

A representation is needed to:

- -store graphs in memory
- -share graphs information

2. Adjacency List

Easy to store AND efficient!

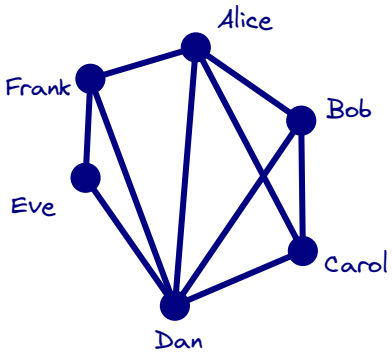


Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

2. Adjacency List

Easy to store AND efficient!



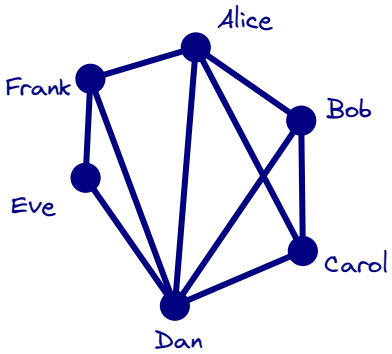
Alice -> Bob, Carol, Dan, Frank
Bob -> Alice, Carol, Dan
Carol -> Alice, Bob, Dan
Dan -> Alice, Bob, Carol, Eve, Frank
Eve -> Dan, Frank
Frank -> Dan, Eve, Alice

Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

2. Adjacency List

Easy to store AND efficient!



Alice -> Bob, Carol, Dan, Frank
Bob -> Alice, Carol, Dan
Carol -> Alice, Bob, Dan
Dan -> Alice, Bob, Carol, Eve, Frank
Eve -> Dan, Frank
Frank -> Dan, Eve, Alice

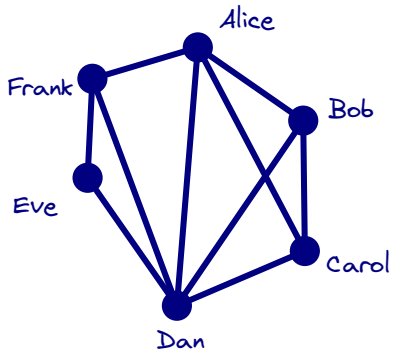
Dan and Frank are friends? Only check Dan's list!
Eve has any friends? Only check Eve's list!

Graph Representations

A representation is needed to:

- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

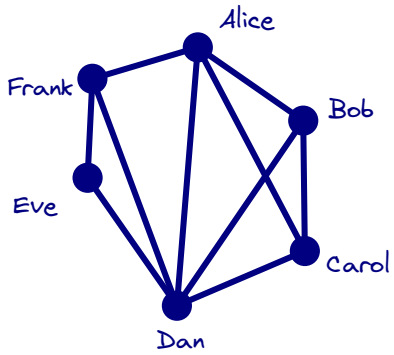


Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

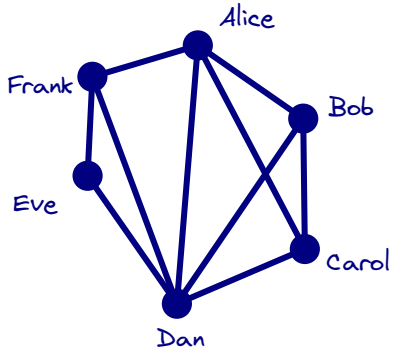


Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

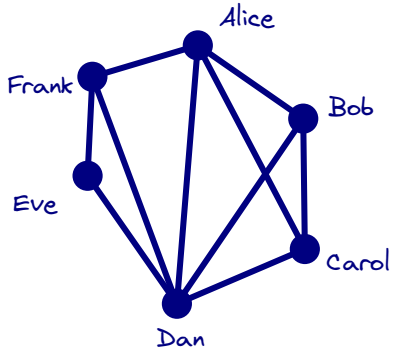
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

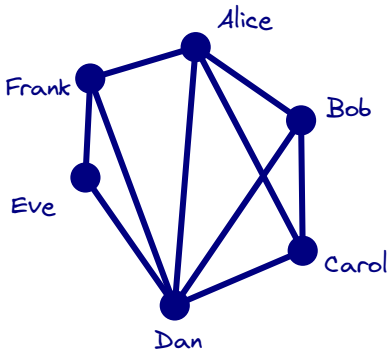
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

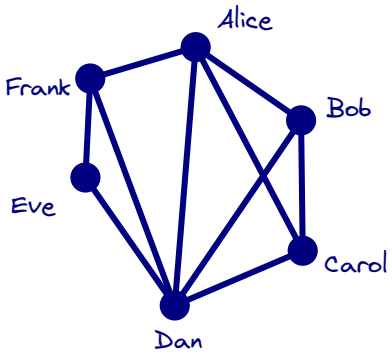
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	1	0	0
C	1	1	0	1	0	0
D	1	1	1	0	1	1
E	0	0	0	1	0	1
F	1	0	0	1	1	0

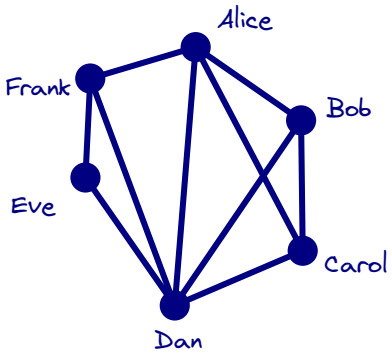
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	1	0	0
C	1	1	0	1	0	0
D	1	1	1	0	1	1
E	0	0	0	1	0	1
F	1	0	0	1	1	0

Dan and Frank are friends? Only check cell with Dan's column and Frank's row!
Eve has any friends? Only check Eve's row!

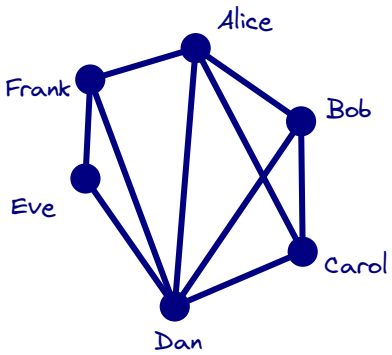
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!



Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	1	0	0
C	1	1	0	1	0	0
D	1	1	1	0	1	1
E	0	0	0	1	0	1
F	1	0	0	1	1	0

Dan and Frank are friends? Only check cell with Dan's column and Frank's row!
Eve has any friends? Only check Eve's row!

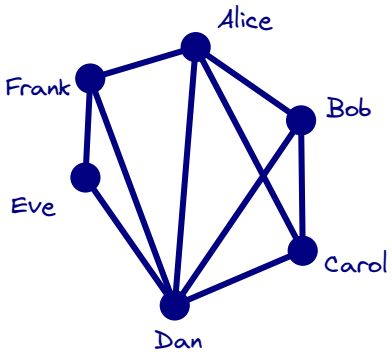
Graph Representations

A representation is needed to:
- -store graphs in memory
- -share graphs information

3. Adjacency Matrix

IDEA: for each possible edge answer YES/NO.

Very efficient but memory costly!

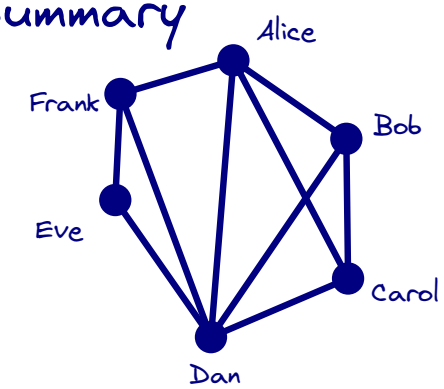


Alice i.f.w. Bob YES
Bob i.f.w. Carol YES
Carol i.f.w. Dan YES
Dan i.f.w. Eve YES
Eve i.f.w. Frank YES
Alice i.f.w. Frank YES
Alice i.f.w. Carol YES
Alice i.f.w. Dan YES
Bob i.f.w. Dan YES
Dan i.f.w. Frank YES
Alice i.f.w. Eve NO
Bob i.f.w. Eve NO
Carol i.f.w. Eve NO
Bob i.f.w. Frank NO
Carol i.f.w. Frank NO

	A	B	C	D	E	F
A	0	1	1	1	0	1
B	1	0	1	1	0	0
C	1	1	0	1	0	0
D	1	1	1	0	1	1
E	0	0	0	1	0	1
F	1	0	0	1	1	0

Dan and Frank are friends? Only check cell with Dan's column and Frank's row!
Eve has any friends? Only check Eve's row!

Graph Representations: Summary

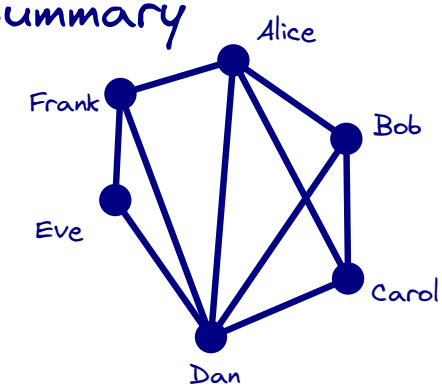


1. Edge List

2. Adjacency List

3. Adjacency Matrix

Graph Representations: Summary



1. Edge List

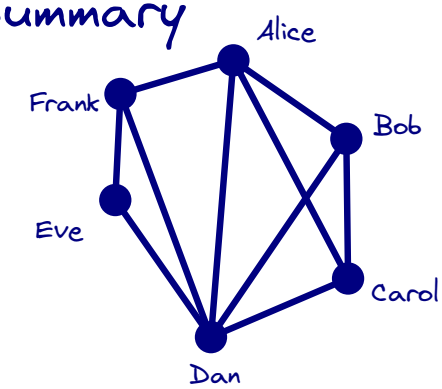
Memory footprint:

$\Theta(|E|)$ we keep all the edges.

2. Adjacency List

3. Adjacency Matrix

Graph Representations: Summary



1. Edge List

Memory footprint:

$\Theta(|E|)$ we keep all the edges.

2. Adjacency List

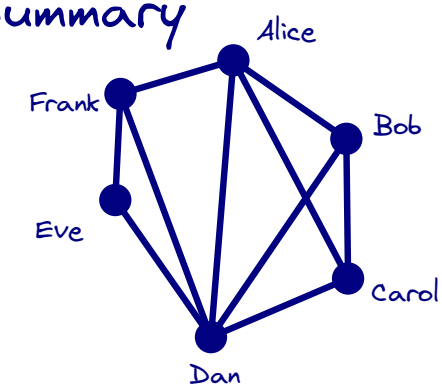
Memory footprint:

$\Theta(|V| + |E|)$ We keep neighbors for each vertex.

3. Adjacency Matrix

Alice -> Bob, Carol, Dan, Frank
Bob -> Alice, Carol, Dan
Carol -> Alice, Bob, Dan
Dan -> Alice, Bob, Carol, Eve, Frank
Eve -> Dan, Frank
Frank -> Dan, Eve, Alice

Graph Representations: Summary



1. Edge List

Memory footprint:

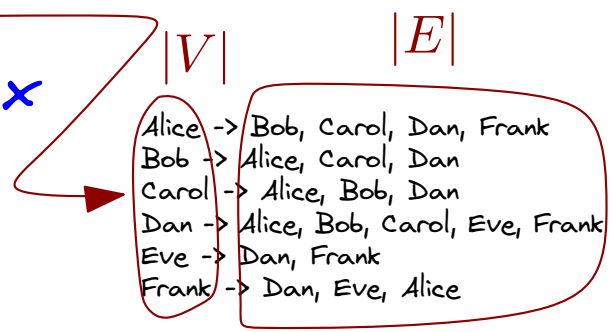
$\Theta(|E|)$ we keep all the edges.

2. Adjacency List

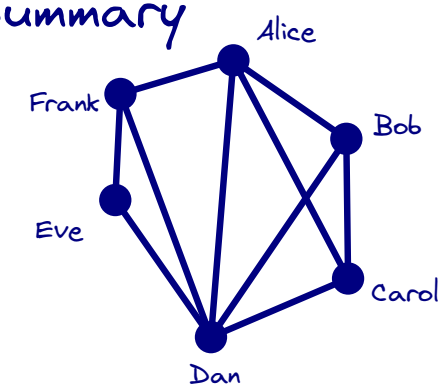
Memory footprint:

$\Theta(|V| + |E|)$ We keep neighbors for each vertex.

3. Adjacency Matrix



Graph Representations: Summary



1. Edge List

Memory footprint:

$\Theta(|E|)$ we keep all the edges.

2. Adjacency List

$$\Theta(|V|) \text{ if } |V| > |E|$$
$$\Theta(|E|) \text{ if } |V| < |E|$$
$$= \max(|V|, |E|)$$

Memory footprint:

$\Theta(|V| + |E|)$ We keep neighbors for each vertex.

3. Adjacency Matrix

$|V|$ $|E|$

Alice -> Bob, Carol, Dan, Frank
Bob -> Alice, Carol, Dan
Carol -> Alice, Bob, Dan
Dan -> Alice, Bob, Carol, Eve, Frank
Eve -> Dan, Frank
Frank -> Dan, Eve, Alice

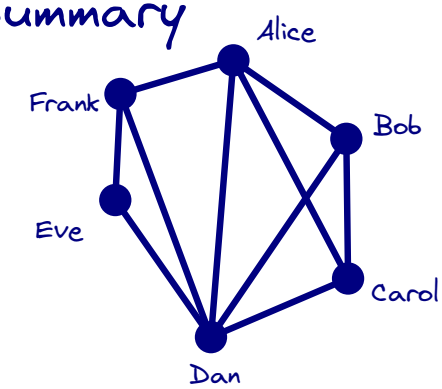
TOPHAT Question

- For which graphs, adjacency list takes $\Theta(|E|)$ memory?

TOPHAT Question

- For which graphs, adjacency list takes $\Theta(|E|)$ memory?
- All "connected graphs" because they have $\Omega(|V|)$ edges.

Graph Representations: Summary



1. Edge List

Memory footprint:

$\Theta(|E|)$ we keep all the edges.

2. Adjacency List

Memory footprint:

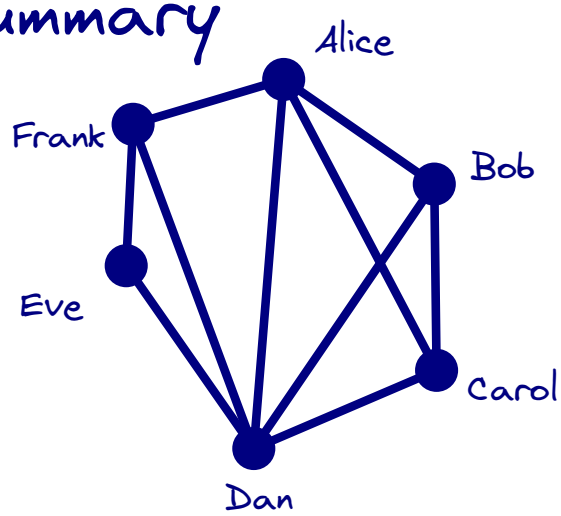
$\Theta(|V| + |E|)$ We keep neighbors for each vertex.

3. Adjacency Matrix

Memory footprint:

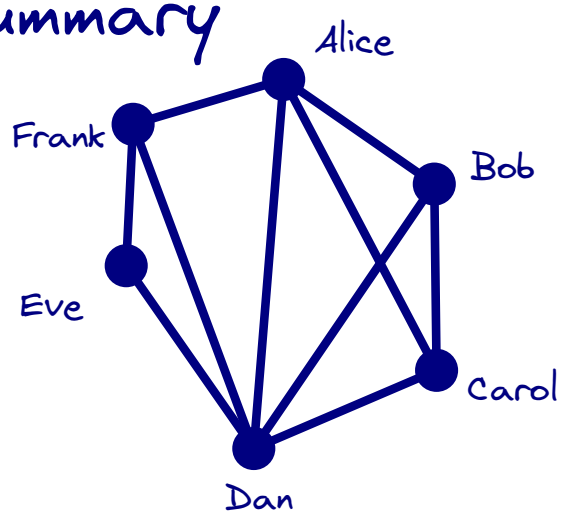
$\Theta(|V|^2)$ We keep a matrix of all possible edges.

Graph Representations: Summary



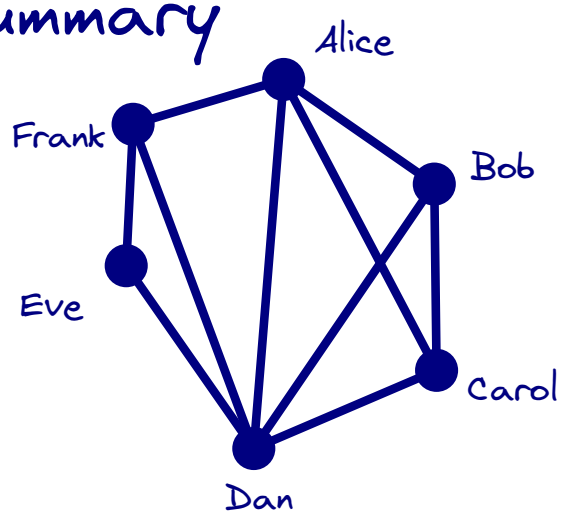
Graph Representations: Summary

1. Edge List



Graph Representations: Summary

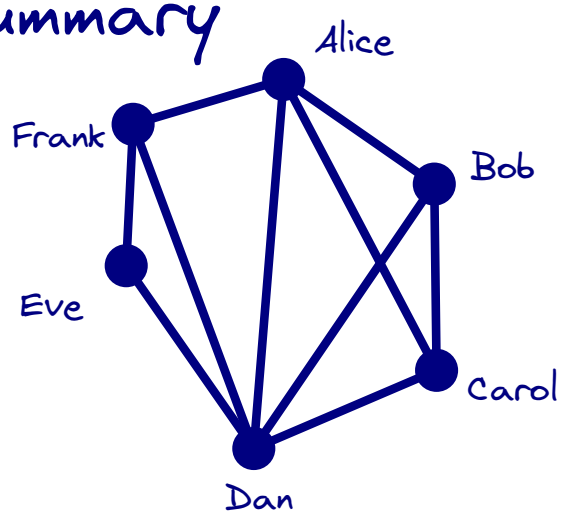
1. Edge List



List all neighbors of a vertex:

$\Theta(|E|)$ we have to go through all the edges.

Graph Representations: Summary



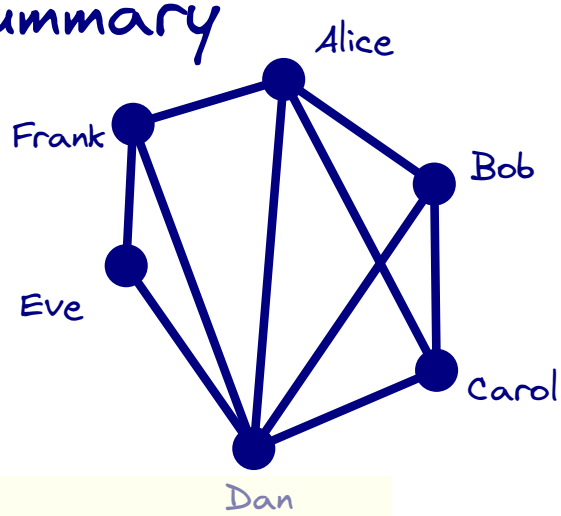
1. Edge List

List all neighbors of a vertex:

$\Theta(|E|)$ we have to go through all the edges.

2. Adjacency List

Graph Representations: Summary



1. Edge List

List all neighbors of a vertex:

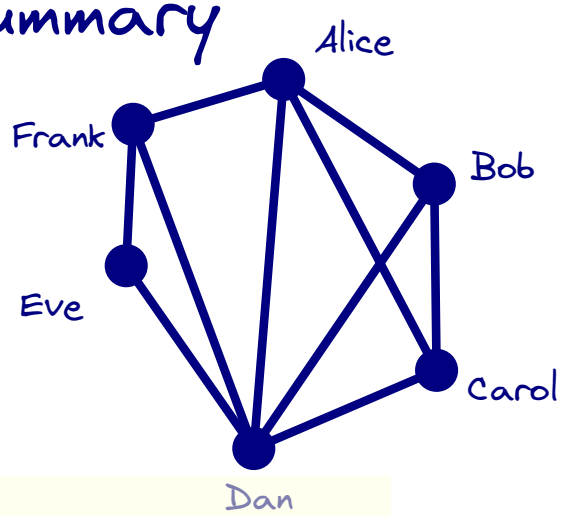
$\Theta(|E|)$ we have to go through all the edges.

2. Adjacency List

List all neighbors of a vertex:

$\Theta(|V|)$ Only need to check one of the lists.

Graph Representations: Summary



1. Edge List

List all neighbors of a vertex:

$\Theta(|E|)$ we have to go through all the edges.

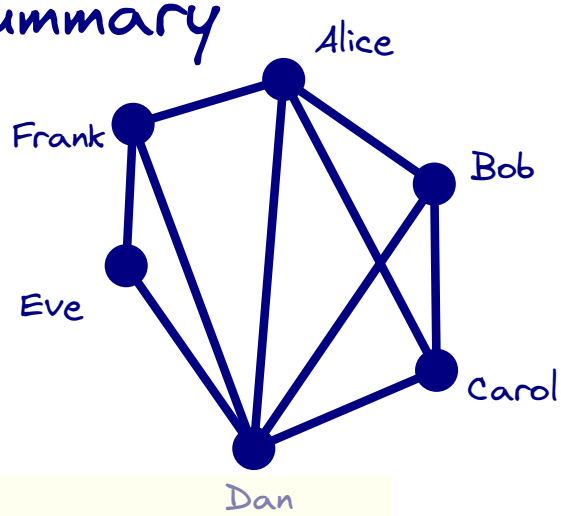
2. Adjacency List

List all neighbors of a vertex:

$\Theta(|V|)$ Only need to check one of the lists.

3. Adjacency Matrix

Graph Representations: Summary



1. Edge List

List all neighbors of a vertex:

$\Theta(|E|)$ we have to go through all the edges.

2. Adjacency List

List all neighbors of a vertex:

$\Theta(|V|)$ Only need to check one of the lists.

3. Adjacency Matrix

List all neighbors of a vertex:

$\Theta(|V|)$ Only need to check one row in the matrix.

Preferred Data Structures/Representations

- Traverse a graph?

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix
- Count Neighbors of a vertex?

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix
- Count Neighbors of a vertex?
- Adjacency List

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix
- Count Neighbors of a vertex?
- Adjacency List
- Operations on a graph with 10^6 vertices?

Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix
- Count Neighbors of a vertex?
- Adjacency List
- Operations on a graph with 10^6 vertices?
- Adjacency List

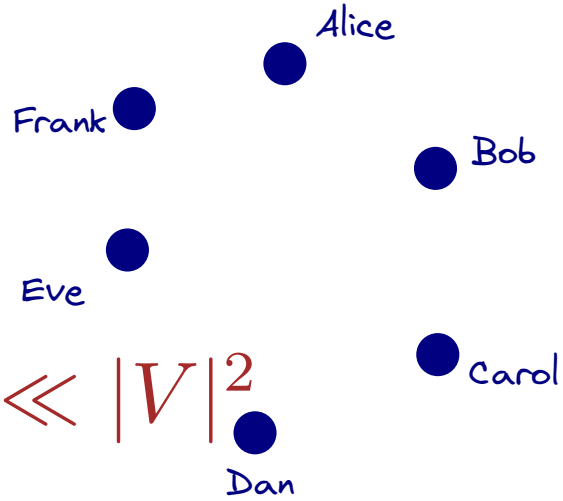
Preferred Data Structures/Representations

- Traverse a graph?
- Adjacency List
- Delete an edge/Insert a new edge?
- Adjacency Matrix
- Count Neighbors of a vertex?
- Adjacency List
- Operations on a graph with 10^6 vertices?
- Adjacency List
- Operations on a sparse graph vs dense graph?

Sparse vs Dense Graphs

Sparse vs Dense Graphs

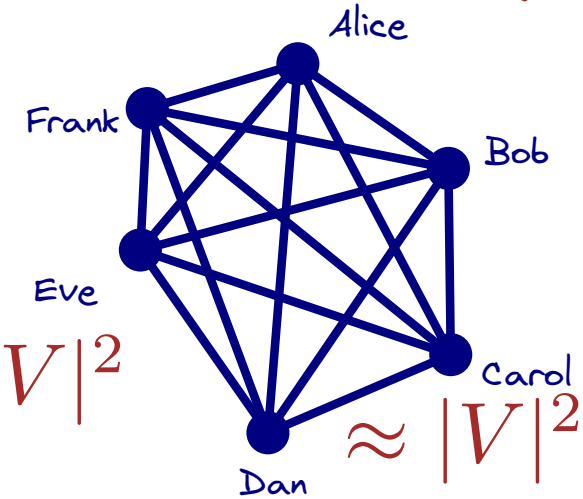
min. number of edges max. number of edges



0

$\ll |V|^2$

$\binom{|V|}{2} \approx |V|^2$



$\approx |V|^2$

SPARSE

twitter: 450 million vertices with 40-50 followers avg.

Facebook: 2.9 billion vertices with 190 friends avg.

DENSE

crypto-currency graph is complete.