MATH 322 – Graph Theory
Fall Term 2021

Notes for Lecture 2

Tuesday, September 7

Definition of the notion of 'graph'

A *graph* G is an ordered pair $\big(V(G), E(G)\big)$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* of G; singular number of the word is *vertex*),
- and where $E(G)$ is a subset of the set of 2-element subsets of $V(G)$ (or in other words, it is a set of unordered pairs of elements from $V(G)$). The elements of $E(G)$ are called the *edges* of G.

Definition of the notion of 'graph'

A *graph* G is an ordered pair $\big(V(G), E(G)\big)$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* of G; singular number of the word is *vertex*),
- and where $E(G)$ is a subset of the set of 2-element subsets of $V(G)$ (or in other words, it is a set of unordered pairs of elements from $V(G)$). The elements of $E(G)$ are called the *edges* of G.

Recall that the vertices of a graph G are also called the *nodes* of G.

Definition of the notion of 'graph'

A *graph* G is an ordered pair $\big(V(G), E(G)\big)$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* of G; singular number of the word is *vertex*),
- and where $E(G)$ is a subset of the set of 2-element subsets of $V(G)$ (or in other words, it is a set of unordered pairs of elements from $V(G)$). The elements of $E(G)$ are called the *edges* of G.

Recall that the vertices of a graph G are also called the *nodes* of G.

Terminology. If e is an edge of G (that is, $e \in E(G)$), then according to our definition there exist x, y in $V(G)$ such that $e = \{x, y\}$. Then we call these vertices x, y the *endvertices* of e, or the *endpoints* or e, or, even simpler, the *ends* of e.

Real-life examples: Social Media Platforms

Example 1: Facebook! We can think of Facebook as a graph:

- our set of vertices are the individual Facebook users (or individual accounts);
- two users/vertices are connected by an edge if they are Facebook friends.

Real-life examples: Social Media Platforms

Example 1: Facebook! We can think of Facebook as a graph:

- our set of vertices are the individual Facebook users (or individual accounts);
- two users/vertices are connected by an edge if they are Facebook friends.

Example 2: Twitter! Similarly, we could view the Twitter platform as a graph:

- the set of vertices here are the different accounts;
- two accounts/vertices are connected by an edge if one of them follows the other.

Facebook vs Twitter?

Question: Do both constructions here work equally well?

That is, do they **both** fit the definition that we gave above, or is it more natural to say that in one of them we are ‘suppressing’ some information that we have?

Reminder: Social Media Platforms viewed as Graphs

Example 1: Facebook! We can think of Facebook as a graph:

- our set of vertices are the individual Facebook users (or individual accounts);
- two users/vertices are connected by an edge if they are Facebook friends.

Example 2: Twitter! Similarly, we could view the Twitter platform as a graph:

- the set of vertices here are the different accounts;
- two accounts/vertices are connected by an edge if one of them follows the other.

Remark. As we discussed last time, the endvertices of an edge in the 'Twitter graph' are not of the same 'importance'. That is, if the account u follows the account w, and we 'encode' this by the edge $\{u, w\}$ (which is the same as $\{w, u\}$; recall that in a set the order of the elements does not matter), then we lose the information that u follows w (while w may not follow u).

Reminder: Social Media Platforms viewed as Graphs

Example 1: Facebook! We can think of Facebook as a graph:

- our set of vertices are the individual Facebook users (or individual accounts);
- two users/vertices are connected by an edge if they are Facebook friends.

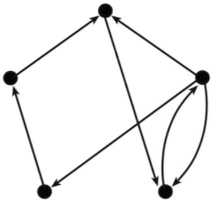
Example 2: Twitter! Similarly, we could view the Twitter platform as a graph:

- the set of vertices here are the different accounts;
- two accounts/vertices are connected by an edge if one of them follows the other.

Remark. As we discussed last time, the endvertices of an edge in the 'Twitter graph' are not of the same 'importance'. That is, if the account u follows the account w, and we 'encode' this by the edge $\{u, w\}$ (which is the same as $\{w, u\}$; recall that in a set the order of the elements does not matter), then we lose the information that u follows w (while w may not follow u).

It appears that we would much better capture the essential information about Twitter if we used the notion of 'directed graph'.

Definition of the notion of 'directed graph' ('digraph' for short)

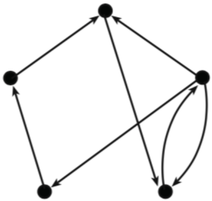


from the HHM book

A *directed graph* G is an ordered pair $\big(V(G), E(G)\big)$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* or *nodes* of G),
- and where $E(G)$ is a subset of **ordered** pairs of elements from $V(G)$). The elements of $E(G)$ are again called the *edges* of G (and each of them now has a specific *orientation* or *direction*).

Definition of the notion of 'directed graph' ('digraph' for short)



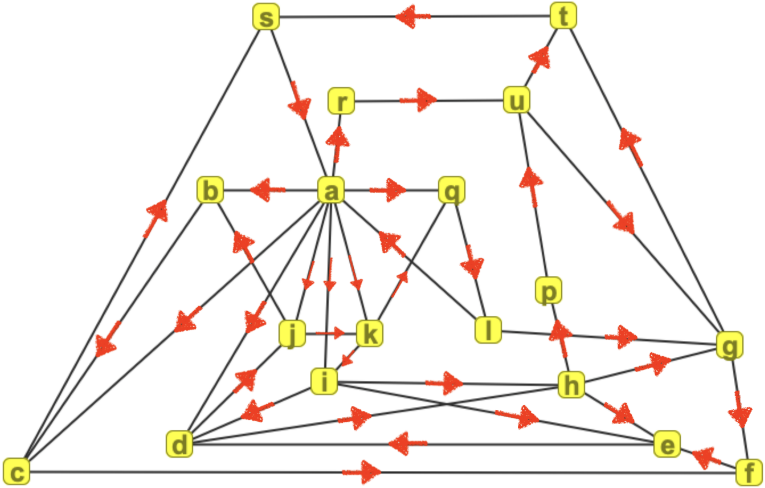
from the HHM book

A *directed graph* G is an ordered pair $\big(V(G), E(G)\big)$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* or *nodes* of G),
- and where $E(G)$ is a subset of **ordered** pairs of elements from $V(G)$). The elements of $E(G)$ are again called the *edges* of G (and each of them now has a specific *orientation* or *direction*).

More about these later in the term....

One more example of a digraph



from the Final Exam - Fall 2020

For easy reference, let us also define the notion of 'multigraph'.

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\{1,2,5,6,7,13,14\} \cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\}$$
$$=$$

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\begin{aligned}\{1,2,5,6,7,13,14\} &\cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\}\\ &= \{1,2,5,6,7,13,14, \qquad\qquad\qquad\qquad \},\end{aligned}$$

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\{1,2,5,6,7,13,14\} \cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\}$$
$$= \{1,2,5,6,7,13,14,\ 2,3,4,5,8,12,14,15, \qquad \},$$

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\begin{aligned}\{1,2,5,6,7,13,14\} \cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\} \\ = \{1,2,5,6,7,13,14,\ 2,3,4,5,8,12,14,15,\ 1,2,8\},\end{aligned}$$

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\begin{aligned}\{1,2,5,6,7,13,14\} \cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\} \\ = \{1,2,5,6,7,13,14,\ 2,3,4,5,8,12,14,15,\ 1,2,8\},\end{aligned}$$

and then of course we should realise that this coincides with the set $\{1,2,3,4,5,6,7,8,12,13,14,15\}$.

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

Recall that in a set each element counts once towards the size/cardinality of the set. Thus, in order to keep our notation as clear as possible, in most instances we DON'T write down any element of the set repeated times.

For example, if we take the union of two or more sets, and we do everything 'mechanically' at first (say, without paying much attention to what we're doing and without trying to remember any of our previous steps), we could write that

$$\begin{aligned}\{1,2,5,6,7,13,14\} \cup \{2,3,4,5,8,12,14,15\} \cup \{1,2,8\} \\ = \{1,2,5,6,7,13,14,\ 2,3,4,5,8,12,14,15,\ 1,2,8\},\end{aligned}$$

and then of course we should realise that this coincides with the set $\{1,2,3,4,5,6,7,8,12,13,14,15\}$.

Sometimes, for simplicity, we even keep the repetitions when describing a set, even though they don't matter at the end: e.g. the set $\left\{\frac{p}{q} : p,q \in \mathbb{Z},\ 2 \leqslant p,q \leqslant 6\right\}$ *contains 17 rational numbers, but some of them are described twice or more in our notation.*

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

On the other hand, a *multiset* is a construction that 'mimics' that of a set, but in which repetition of elements is allowed and does count.

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

On the other hand, a *multiset* is a construction that 'mimics' that of a set, but in which repetition of elements is allowed and does count.

E.g. if we now view the objects

$$\{1, 2, 5, 6, 7, 13, 14,\ \ 2, 3, 4, 5, 8, 12, 14, 15,\ \ 1, 2, 8\}$$
$$= \{1, 1, 2, 2, 2, 3, 4, 5, 5, 6, 7, 8, 8, 12, 13, 14, 14, 15\}$$

and

$$\{1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15\}$$

as multisets, <u>then they are different objects!</u>

Sets vs Multisets / Graphs vs Multigraphs

A bit of Set Theory

On the other hand, a *multiset* is a construction that 'mimics' that of a set, but in which repetition of elements is allowed and does count.

E.g. if we now view the objects

$$\{1, 2, 5, 6, 7, 13, 14,\ \ 2, 3, 4, 5, 8, 12, 14, 15,\ \ 1, 2, 8\}$$
$$= \{1, 1, 2, 2, 2, 3, 4, 5, 5, 6, 7, 8, 8, 12, 13, 14, 14, 15\}$$

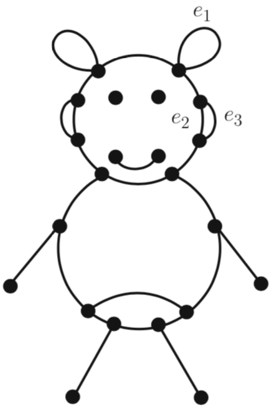
and

$$\{1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15\}$$

as multisets, then they are different objects!

With this terminology at hand...

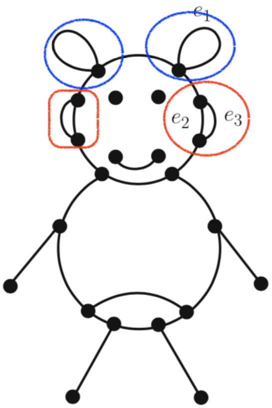
Definition of a ‘multigraph’



A *multigraph* G is an ordered pair $(V(G), E(G))$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* or *nodes* of G),
- and where $E(G)$ is a **multiset** whose elements are taken from the set of 2-element and 1-element subsets of $V(G)$. The elements of $E(G)$ are called the *edges* of G

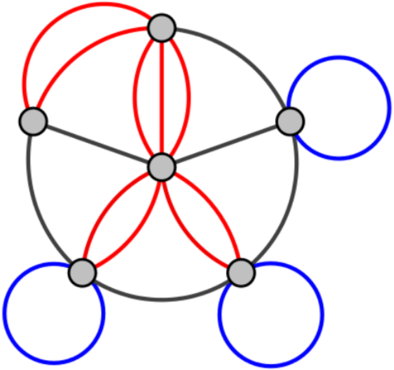
Definition of a ‘multigraph’



A *multigraph* G is an ordered pair $(V(G), E(G))$,

- where $V(G)$ is a non-empty set (whose elements are called the *vertices* or *nodes* of G),
- and where $E(G)$ is a **multiset** whose elements are taken from the set of 2-element and 1-element subsets of $V(G)$. The elements of $E(G)$ are called the *edges* of G (and now we can also have ‘repeated’ edges connecting the same pair of vertices, which we call groups of *multiple edges* or *parallel edges*, as well as *loops* (we can even have *multiple (or parallel) loops* at the same vertex)).

Some more examples of multigraphs



ATTENTION to the conventions

Terminology NOT Universally Agreed On

- Some authors and some books use the term **'graph'** even when multiple edges and/or loops are included (thus they use the term 'graph' in place of the term 'multigraph').
- Then, to distinguish with the important family of cases where neither multiple edges nor loops are allowed, they use the term **'simple graph'**.

ATTENTION to the conventions

Terminology NOT Universally Agreed On

• Some authors and some books use the term **'graph'** even when multiple edges and/or loops are included (thus they use the term 'graph' in place of the term 'multigraph').

• Then, to distinguish with the important family of cases where neither multiple edges nor loops are allowed, they use the term **'simple graph'**.

Out of the list of recommended books in this course's syllabus:

- Balakrishnan and Ranganathan's book uses the term **'graph'** even when multiple edges and/or loops are included, and uses the term **'simple graph'** when these are not allowed.
- Bollobas' book, Marcus' book and Wallis' book use the term **'multigraph'** when multiple edges and/or loops are allowed/included, and the term **'graph'** when they are not allowed.
- Finally, Harris, Hirst and Mossinghoff's book **does not allow** multiple edges or loops in a **'graph'**, uses the term **'multigraph'** when multiple edges are allowed/included, but at the same time **does not allow** loops in a **'multigraph'**, and uses the term **'pseudograph'** when loops are also allowed.

ATTENTION to the conventions

Terminology NOT Universally Agreed On

• Some authors and some books use the term **'graph'** even when multiple edges and/or loops are included (thus they use the term 'graph' in place of the term 'multigraph').

• Then, to distinguish with the important family of cases where neither multiple edges nor loops are allowed, they use the term **'simple graph'**.

Out of the list of recommended books in this course's syllabus:

- Balakrishnan and Ranganathan's book uses the term **'graph'** even when multiple edges and/or loops are included, and uses the term **'simple graph'** when these are not allowed.
- Bollobas' book, Marcus' book and Wallis' book use the term **'multigraph'** when multiple edges and/or loops are allowed/included, and the term **'graph'** when they are not allowed.
- Finally, Harris, Hirst and Mossinghoff's book **does not allow** multiple edges or loops in a **'graph'**, uses the term **'multigraph'** when multiple edges are allowed/included, but at the same time **does not allow** loops in a **'multigraph'**, and uses the term **'pseudograph'** when loops are also allowed.

Convention for this course: we will use the term **'multigraph'** when multiple edges and/or loops are allowed/included, and the term **'graph'** when they are not allowed (and our primary focus will be on graphs).

Back to Graphs now

Basic Terminology

Terms we have already seen

- graph
- vertex (or node)
- edge
- labelled
- unlabelled
- multigraph
- multiple edges (or parallel edges)
- loops
- connected

More terms to see today

Basic Terminology

Terms we have already seen

- graph
- vertex (or node)
- edge
- labelled
- unlabelled
- multigraph
- multiple edges (or parallel edges)
- loops
- connected

More terms to see today

- adjacency
- incidence

Basic Terminology

Terms we have already seen

- graph
- vertex (or node)
- edge
- labelled
- unlabelled
- multigraph
- multiple edges (or parallel edges)
- loops
- connected

More terms to see today

- adjacency
- incidence
- order of a graph
- size of a graph
- finite graph

Basic Terminology

Terms we have already seen

- graph
- vertex (or node)
- edge
- labelled
- unlabelled
- multigraph
- multiple edges (or parallel edges)
- loops
- connected

More terms to see today

- adjacency
- incidence
- order of a graph
- size of a graph
- finite graph
- neighbourhood of a vertex
- degree of a vertex

Basic Terminology

Terms we have already seen

- graph
- vertex (or node)
- edge
- labelled
- unlabelled
- multigraph
- multiple edges (or parallel edges)
- loops
- connected

More terms to see today

- adjacency
- incidence
- order of a graph
- size of a graph
- finite graph
- neighbourhood of a vertex
- degree of a vertex
- subgraph
- induced subgraph
- path

Labelled vs Unlabelled / Order and Size

Recall that we can represent a graph $G = (V, E)$ as a diagram by drawing its vertices as bullet points (or small circles), and by connecting two of these points with a straight line segment, or a more general curve, exactly when there is an edge in G whose endvertices correspond to these points.

If we also label the bullet points/vertices in the diagram, then we call the resulting diagram (which we will also refer to as the graph G for simplicity) *labelled*. Otherwise we call it *unlabelled*.

Labelled vs Unlabelled / Order and Size

Recall that we can represent a graph $G = (V, E)$ as a diagram by drawing its vertices as bullet points (or small circles), and by connecting two of these points with a straight line segment, or a more general curve, exactly when there is an edge in G whose endvertices correspond to these points.

If we also label the bullet points/vertices in the diagram, then we call the resulting diagram (which we will also refer to as the graph G for simplicity) *labelled*. Otherwise we call it *unlabelled*.

Definition. The size/cardinality of the set V of vertices of G is called the *order* of G, while the cardinality of the set E of edges of G is called the *size* of G.

If the order of G is finite (that is, if it is a natural number), we say that G is a *finite graph*. In this course we will primarily focus on finite graphs.

Adjacent or neighbouring vertices

Let $G = (V, E)$ be a graph.

- Two different vertices $x, y \in V$ are called *adjacent* or *neighbouring* if $\{u, v\} \in E$, that is, if there is an edge e in E with endvertices x and y.

Adjacent or neighbouring vertices

Let $G = (V, E)$ be a graph.

- Two different vertices $x, y \in V$ are called *adjacent* or *neighbouring* if $\{u, v\} \in E$, that is, if there is an edge e in E with endvertices x and y.
- We say that a vertex x is *incident* with an edge e of G, if x is one of the endvertices of e.

Adjacent or neighbouring vertices

Let $G = (V, E)$ be a graph.

- Two different vertices $x, y \in V$ are called *adjacent* or *neighbouring* if $\{u, v\} \in E$, that is, if there is an edge e in E with endvertices x and y.
- We say that a vertex x is *incident* with an edge e of G, if x is one of the endvertices of e.
- Two different edges $e_1, e_2 \in E$ are called *adjacent* if they have a common endvertex (that is, if there exists $z \in V$ such that z is incident both with e_1 and with e_2).

Adjacent or neighbouring vertices

Let $G = (V, E)$ be a graph.

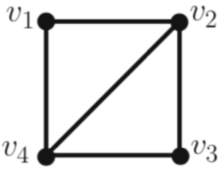
- Two different vertices $x, y \in V$ are called *adjacent* or *neighbouring* if $\{u, v\} \in E$, that is, if there is an edge e in E with endvertices x and y.

- We say that a vertex x is *incident* with an edge e of G, if x is one of the endvertices of e.

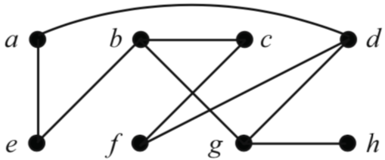
- Two different edges $e_1, e_2 \in E$ are called *adjacent* if they have a common endvertex (that is, if there exists $z \in V$ such that z is incident both with e_1 and with e_2).

- A vertex x of G that has no neighbours in G is called an *isolated vertex* of G (note that this happens if we have that, for every $e \in E$, $x \notin e$ (or, using the terminology above, for every $e \in E$, x is NOT incident with e)).

Back to examples from Lecture 1



Graph G_1



Graph G_2

Some Practice Questions: (i) In each of the two graphs, find pairs of vertices which are NOT adjacent.

(ii) Find pairs of edges which are NOT adjacent.

(iii) Does any of these graphs have isolated vertices?

Adjacency Matrix of a Graph

We can encode in a very efficient way all the pairs of neighbours in a graph $G = (V, E)$, as well as plenty more information as we will soon see, by writing down the *adjacency matrix* of G:

if the set of vertices $V(G)$ of G is, say, the set $\{v_1, v_2, v_3, \dots, v_n\}$, and thus the order of G is n, then this is an $n \times n$ square matrix with only 0 or 1 entries, such that

- the (i, j)-th entry is equal to 1 if $v_i v_j$ is an edge of G (in other words, if $\{v_i, v_j\} \in E$),
- the (i, j)-th entry is equal to 0 otherwise.

Adjacency Matrix of a Graph

We can encode in a very efficient way all the pairs of neighbours in a graph $G = (V, E)$, as well as plenty more information as we will soon see, by writing down the *adjacency matrix* of G:

if the set of vertices $V(G)$ of G is, say, the set $\{v_1, v_2, v_3, \ldots, v_n\}$, and thus the order of G is n, then this is an $n \times n$ square matrix with only 0 or 1 entries, such that

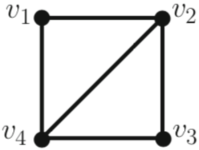
- the (i, j)-th entry is equal to 1 if $v_i v_j$ is an edge of G (in other words, if $\{v_i, v_j\} \in E$),
- the (i, j)-th entry is equal to 0 otherwise.

Note. From the way we define the adjacency matrix, we can conclude that

- it is a symmetric matrix (remember that the edge $v_i v_j$ can also be written as $v_j v_i$);
- its diagonal entries are all equal to 0.

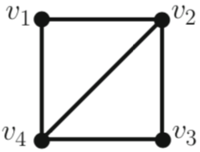
Example: The adjacency matrix of G_1

Recall G_1:



Example: The adjacency matrix of G_1

Recall G_1:

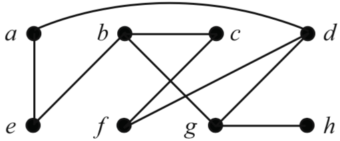


Its adjacency matrix is

$$\begin{array}{c c}
 & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}
\end{array}.$$

Example: The adjacency matrix of G_2

Recall G_2:



Its adjacency matrix is

$$
\begin{array}{c c}
 & \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\end{array}.
$$

Neighbourhood and degree of a vertex

Let $G = (V, E)$ be a graph, and let $x \in V$.
The *(open) neighbourhood* $N(x)$ of x in G is the set of all neighbours of x in G, that is,

$$N(x) = \{y \in V : xy \in E\}.$$

Neighbourhood and degree of a vertex

Let $G = (V, E)$ be a graph, and let $x \in V$.
The *(open) neighbourhood* $N(x)$ of x in G is the set of all neighbours of x in G, that is,

$$N(x) = \{y \in V : xy \in E\}.$$

Important Definition. The *degree* of a vertex x of G is the size/cardinality of the neighbourhood $N(x)$ of x. In other words, it is the total number of neighbours of x.

Neighbourhood and degree of a vertex

Let $G = (V, E)$ be a graph, and let $x \in V$.
The *(open) neighbourhood* $N(x)$ of x in G is the set of all neighbours of x in G, that is,

$$N(x) = \{y \in V : xy \in E\}.$$

Important Definition. The *degree* of a vertex x of G is the size/cardinality of the neighbourhood $N(x)$ of x. In other words, it is the total number of neighbours of x.

Observe that any isolated vertex of G has degree 0 (since its neighbourhood is going to be the empty set).

Neighbourhood and degree of a vertex

Let $G = (V, E)$ be a graph, and let $x \in V$.
The *(open) neighbourhood* $N(x)$ of x in G is the set of all neighbours of x in G, that is,

$$N(x) = \{y \in V : xy \in E\}.$$

Important Definition. The *degree* of a vertex x of G is the size/cardinality of the neighbourhood $N(x)$ of x. In other words, it is the total number of neighbours of x.

Observe that any isolated vertex of G has degree 0 (since its neighbourhood is going to be the empty set).

Very Important Remark. It is very easy to find the degree of any vertex x of G by looking at the adjacency matrix of G.
(How? What do you need to look for?)

Subgraphs

Let $G = (V, E)$ be a graph.

Subgraphs

Let $G = (V, E)$ be a graph.

Definition. A subgraph H of G is an ordered pair $\left(V', E'\right)$

- where $\emptyset \neq V' \subseteq V$ (that is, V' is a non-empty subset of V),
- and where $E' \subseteq E$ with every edge $e \in E'$ having both endvertices in V'.

In this case, we write $H \subseteq G$.

Subgraphs

Let $G = (V, E)$ be a graph.

Definition. A subgraph H of G is an ordered pair $\big(V', E'\big)$

- where $\emptyset \neq V' \subseteq V$ (that is, V' is a non-empty subset of V),
- and where $E' \subseteq E$ with every edge $e \in E'$ having both endvertices in V'.

In this case, we write $H \subseteq G$.

Definition. If $H = \big(V', E'\big)$ is a subgraph of G, and E' contains all the possible edges of G which have both endvertices in V', then we say that H is the subgraph of G that is *induced* or *spanned* by V'.

We denote this induced subgraph by $G[V']$.

Subgraphs

Let $G = (V, E)$ be a graph.

Definition. A subgraph H of G is an ordered pair (V', E')

- where $\emptyset \neq V' \subseteq V$ (that is, V' is a non-empty subset of V),
- and where $E' \subseteq E$ with every edge $e \in E'$ having both endvertices in V'.

In this case, we write $H \subseteq G$.

Definition. If $H = (V', E')$ is a subgraph of G, and E' contains all the possible edges of G which have both endvertices in V', then we say that H is the subgraph of G that is *induced* or *spanned* by V'.

We denote this induced subgraph by $G[V']$.

Useful to note: The induced subgraph $G[V']$ is the 'largest' subgraph of G out of those subgraphs with vertex set V'. 'Largest' here means that it has the most edges possible (also, any other subgraph of G with vertex set V' will be a subgraph of $G[V']$ as well).

A very simple type of graphs

Definition. A path P is a graph of the form

$$\Big(\{x_0, x_1, x_2, \ldots, x_l\},\ \{x_0x_1, x_1x_2, \ldots, x_{l-1}x_l\}\Big)$$

where l is an integer $\geqslant 1$.

A very simple type of graphs

Definition. A path P is a graph of the form

$$\Big(\{x_0, x_1, x_2, \ldots, x_l\},\ \{x_0x_1, x_1x_2, \ldots, x_{l-1}x_l\}\Big)$$

where l is an integer $\geqslant 1$.

The number l is called the *length* of the path P (note that it is also the number of edges of P, that is, it is equal to the size of P).

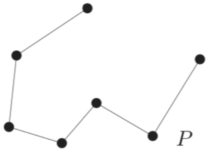
A very simple type of graphs

Definition. A path P is a graph of the form

$$\Big(\{x_0, x_1, x_2, \ldots, x_l\},\ \{x_0x_1, x_1x_2, \ldots, x_{l-1}x_l\}\Big)$$

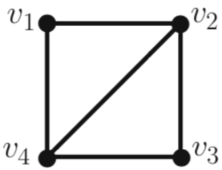
where l is an integer $\geqslant 1$.

The number l is called the *length* of the path P (note that it is also the number of edges of P, that is, it is equal to the size of P).

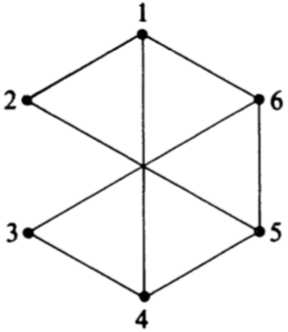


A path P on 7 vertices, thus of length 6

Back to examples from Lecture 1

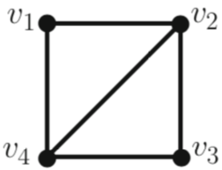


Graph G_1

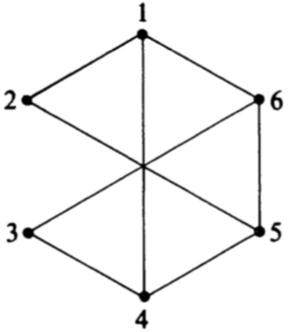


Graph G_3

Back to examples from Lecture 1



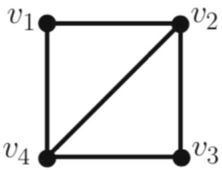
Graph G_1



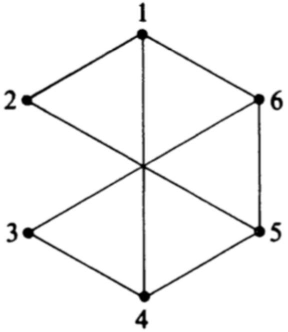
Graph G_3

Question 1. Can we view G_1 as a subgraph of G_3? (here of course you do need to and you can relabel the vertices of G_1 using four of the numbers from $\{1, 2, 3, 4, 5, 6\}$)

Back to examples from Lecture 1



Graph G_1

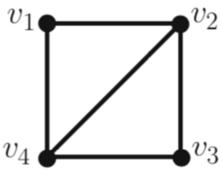


Graph G_3

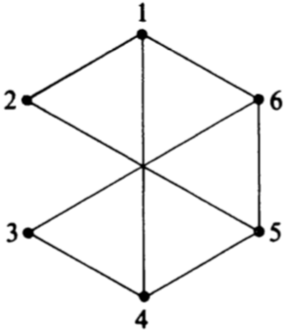
Question 1. Can we view G_1 as a subgraph of G_3? (here of course you do need to and you can relabel the vertices of G_1 using four of the numbers from $\{1, 2, 3, 4, 5, 6\}$)

Question 2. How many subgraphs of G_3 do we have on the vertices $\{2, 3, 4, 5\}$? Can you draw them? What is the induced subgraph here?

Back to examples from Lecture 1



Graph G_1



Graph G_3

Question 1. Can we view G_1 as a subgraph of G_3? (here of course you do need to and you can relabel the vertices of G_1 using four of the numbers from $\{1, 2, 3, 4, 5, 6\}$)

Question 2. How many subgraphs of G_3 do we have on the vertices $\{2, 3, 4, 5\}$? Can you draw them? What is the induced subgraph here?

Question 3. What is the maximum length of a path in G_3? How many paths can you find with this length?

A final reminder: Proper definition of ‘connected graph’

Definition. A graph $G = (V, E)$ is called *connected* if, for every two different vertices $x, y \in V$, there is a path from x to y (or equivalently from y to x) in G (in other words, there is a subgraph P of G which is a path from x to y).

A final reminder: Proper definition of 'connected graph'

Definition. A graph $G = (V, E)$ is called *connected* if, for every two different vertices $x, y \in V$, there is a path from x to y (or equivalently from y to x) in G (in other words, there is a subgraph P of G which is a path from x to y).

A graph G that is not connected is called *disconnected*.

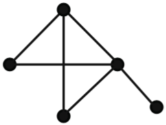
A final reminder: Proper definition of 'connected graph'

Definition. A graph $G = (V, E)$ is called *connected* if, for every two different vertices $x, y \in V$, there is a path from x to y (or equivalently from y to x) in G (in other words, there is a subgraph P of G which is a path from x to y).

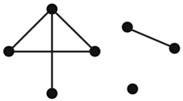
A graph G that is not connected is called *disconnected*.

Remark. Clearly a graph G that contains isolated vertices (and has at least two vertices) will be disconnected. However, we can also find disconnected graphs that do not have any isolated vertices (see examples on next slide).

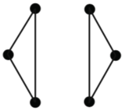
Pics from the HHM book



Connected graph

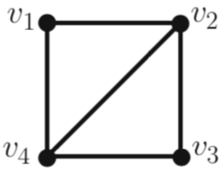


Disconnected; has isolated vertex

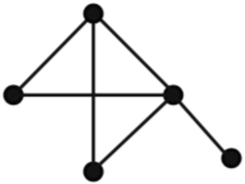


Disconnected; no isolated vertices

For the end, a slightly trickier question



Graph G_1



Graph G_4

Can you view G_1 as a subgraph of G_4 or not? (That is, after you appropriately label the vertices of G_4)