Lecture 1.1, MATH-42021/52021 Graph Theory and Combinatorics.

Artem Zvavitch

Department of Mathematical Sciences, Kent State University

June, 2016.

There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

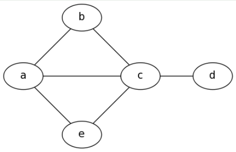
"Introduction" to graphs.

There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.

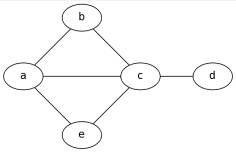
There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.



There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

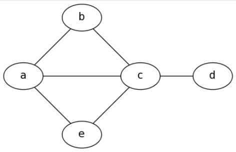
Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.



$V = \{a, b, c, d, e\}$ and $E = \{(a, b), (b, c), (a, c), (e, c), (e, a), (d, c)\}$.

There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.

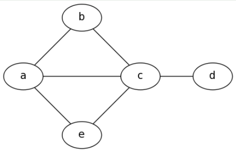


$V = \{a, b, c, d, e\}$ and $E = \{(a, b), (b, c), (a, c), (e, c), (e, a), (d, c)\}$.

- Note when we define the edge the order of vertices does not meter i.e. (a, b) is the same age as (b, a) (yes this comes from the fact that graph is undirected, we do not care about directions here).

There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.

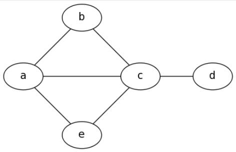


$V = \{a, b, c, d, e\}$ and $E = \{(a, b), (b, c), (a, c), (e, c), (e, a), (d, c)\}$.

- Note when we define the edge the order of vertices does not meter i.e. (a, b) is the same age as (b, a) (yes this comes from the fact that graph is undirected, we do not care about directions here).
- If (a, b) is an edge. We call a, b - ends of the edge (a, b). We agree that ends of an edge must be different! (i.e. $a \neq b$).

There are different definitions of Graphs and a number of different and very interesting generalizations. We will start with the most basic and classical one, (or better to say simple, undirected graph) we will be using it through the class.

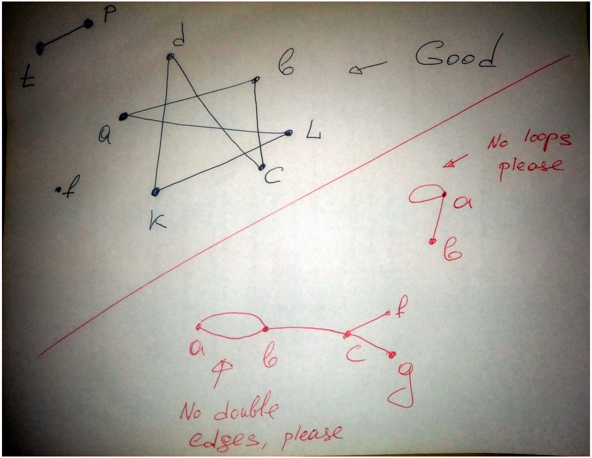
Graph is a pair $G = (V, E)$, a set V is a set of **vertices** and a set E is a set of **edges**, joining different pairs of distinct vertices.



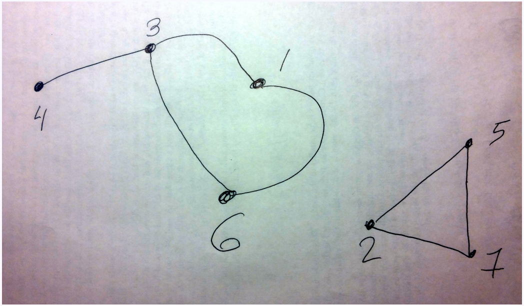
$V = \{a, b, c, d, e\}$ and $E = \{(a, b), (b, c), (a, c), (e, c), (e, a), (d, c)\}$.

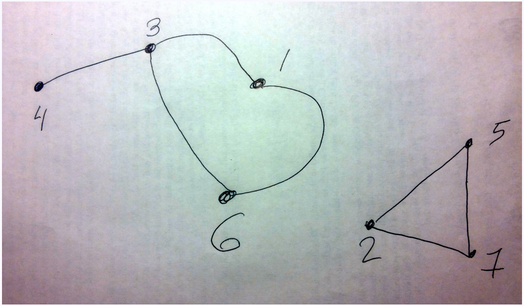
- Note when we define the edge the order of vertices does not meter i.e. (a, b) is the same age as (b, a) (yes this comes from the fact that graph is undirected, we do not care about directions here).
- If (a, b) is an edge. We call a, b - ends of the edge (a, b). We agree that ends of an edge must be different! (i.e. $a \neq b$).
- We also agree that there is at most one edge between any two vertices. (yes this comes from "simple")

More Examples.

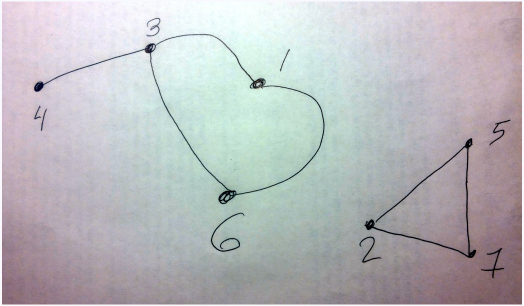


Bit more of definitions



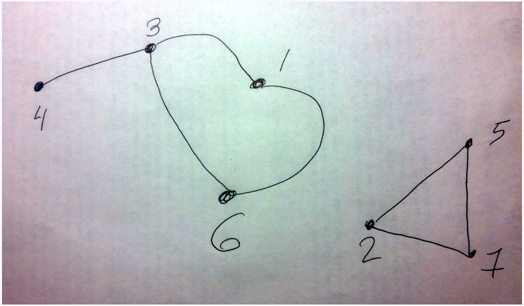


- a and b are adjacent vertices if they are connected by edge (i.e. if they are the end points of an edge, i.e. if $(a, b) \in E$). Example: 2 and 5 are adjacent (because $(2, 5) \in E$), but 4 and 1 are not, because $(4, 1) \notin E$.

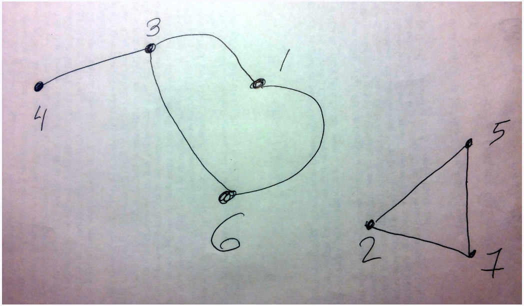


- a and b are adjacent vertices if they are connected by edge (i.e. if they are the end points of an edge, i.e. if $(a, b) \in E$). Example: 2 and 5 are adjacent (because $(2, 5) \in E$), but 4 and 1 are not, because $(4, 1) \notin E$.
- We say that an edge is an incident to a vertex if the vertex is an end point of this edge. Example: $(4, 3)$ is incident to 4 and 3 but not to 2.

Bit more of definitions



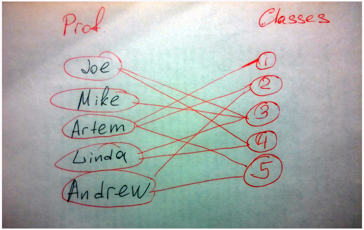
- a and b are adjacent vertices if they are connected by edge (i.e. if they are the end points of an edge, i.e. if $(a, b) \in E$). Example: 2 and 5 are adjacent (because $(2, 5) \in E$), but 4 and 1 are not, because $(4, 1) \notin E$.
- We say that an edge is an incident to a vertex if the vertex is an end point of this edge. Example: $(4, 3)$ is incident to 4 and 3 but not to 2.
- **Path:** is a sequence of *distinct* vertices $(x_1, x_2, \ldots, x_n)$ such that consecutive vertices are adjacent. Example: $(4, 3, 1, 6)$ or $(2, 5, 7)$. Note: $(3, 6, 1, 4)$ not a path. $(4, 3, 1, 6, 3)$ not a path. And there is no pass from 2 to 1.



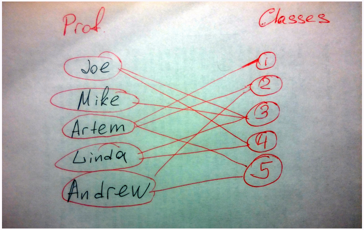
- a and b are adjacent vertices if they are connected by edge (i.e. if they are the end points of an edge, i.e. if $(a,b) \in E$). Example: 2 and 5 are adjacent (because $(2,5) \in E$), but 4 and 1 are not, because $(4,1) \notin E$.
- We say that an edge is an incident to a vertex if the vertex is an end point of this edge. Example: $(4,3)$ is incident to 4 and 3 but not to 2.
- **Path:** is a sequence of *distinct* vertices $(x_1, x_2, \dots, x_n)$ such that consecutive vertices are adjacent. Example: $(4,3,1,6)$ or $(2,5,7)$. Note: $(3,6,1,4)$ not a path. $(4,3,1,6,3)$ not a path. And there is no pass from 2 to 1.
- **Circuit**: is a pass that ends where it starts (i.e. $x_1 = x_n$, yes this is a bit of mix up because we repeated the last vertex, but this is allowed for this very special case and for only those vertices). Example: $(3,1,6,3)$, $(2,5,7,2)$.

Every Spring faculty of Math. Dep. of Aurora State University submit their request for available classes to teach over the summer, there are five people and there are five classes. The problem is to find one-to-one matching of Professors to classes or to show that such matching does not exist (and to ask Prof. to be more reasonable with their requests).

Every Spring faculty of Math. Dep. of Aurora State University submit their request for available classes to teach over the summer, there are five people and there are five classes. The problem is to find one-to-one matching of Professors to classes or to show that such matching does not exist (and to ask Prof. to be more reasonable with their requests). We can represent this situation as a graph:

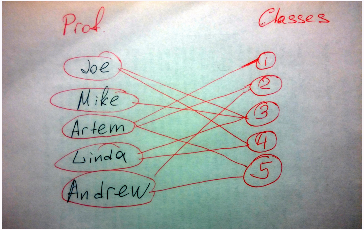


Every Spring faculty of Math. Dep. of Aurora State University submit their request for available classes to teach over the summer, there are five people and there are five classes. The problem is to find one-to-one matching of Professors to classes or to show that such matching does not exist (and to ask Prof. to be more reasonable with their requests). We can represent this situation as a graph:



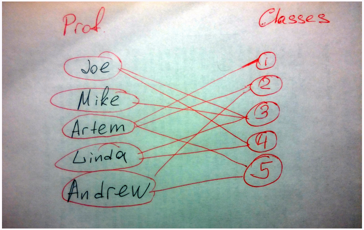
So does it exists a one-to-one matching of Professors to classes in the above graph?

Every Spring faculty of Math. Dep. of Aurora State University submit their request for available classes to teach over the summer, there are five people and there are five classes. The problem is to find one-to-one matching of Professors to classes or to show that such matching does not exist (and to ask Prof. to be more reasonable with their requests). We can represent this situation as a graph:



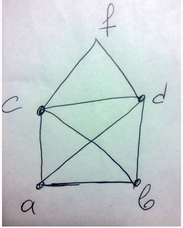
So does it exists a one-to-one matching of Professors to classes in the above graph? Unfortunately, NO. (just Look at what Joe, Linda and Mike want).

Every Spring faculty of Math. Dep. of Aurora State University submit their request for available classes to teach over the summer, there are five people and there are five classes. The problem is to find one-to-one matching of Professors to classes or to show that such matching does not exist (and to ask Prof. to be more reasonable with their requests). We can represent this situation as a graph:



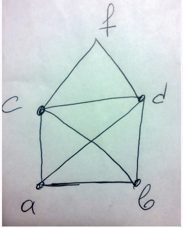
So does it exists a one-to-one matching of Professors to classes in the above graph? Unfortunately, NO. (just Look at what Joe, Linda and Mike want). The above graph is an example of a **bipartite** graph (is a graph whose vertices can be divided into two disjoint sets and such that every edge connects a vertex from one set to another - i.e. for each edge the end points belong to different sets). We will sure talk much more about those guys.

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



Example: Network's vulnerability

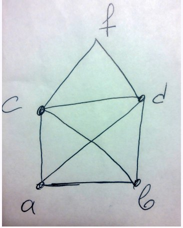
Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



Actually, it is clear that we can remove any edge in this graph and keep the network connected.

Example: Network's vulnerability

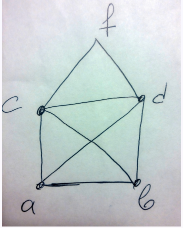
Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex).

Example: Network's vulnerability

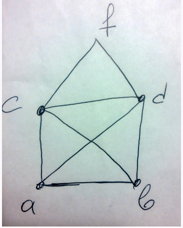
Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices).

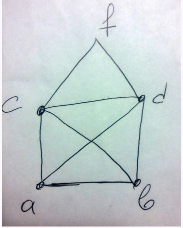
Example: Network's vulnerability

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.

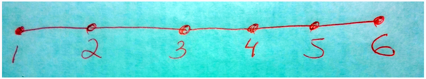


Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices).
We can ask a different question: what is the minimal number of edges we need to *keep* so that the network is still connected?

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



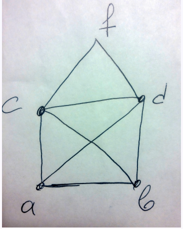
Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices).
We can ask a different question: what is the minimal number of edges we need to *keep* so that the network is still connected? Lets first play with some graphs:



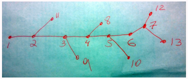
There is no edge we can remove here! Is there something special in the "geometry" of this graph?

Example: Network's vulnerability

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



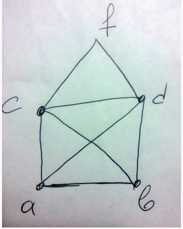
Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices). We can ask a different question: what is the minimal number of edges we need to *keep* so that the network is still connected? Lets first play with some graphs:



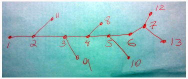
There is nothing we can remove here! Is there something special in the "geometry" of this graph? Yes, it is a tree (a graph in which any two vertices are connected by exactly one path, i.e. connected graph with no circuits - we will talk much more about those guys in the future).

Example: Network's vulnerability

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.



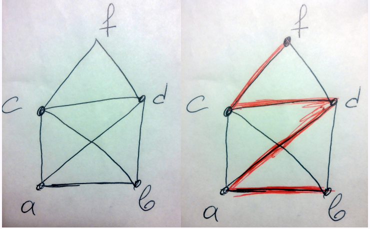
Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices). We can ask a different question: what is the minimal number of edges we need to *keep* so that the network is still connected? Lets first play with some graphs:



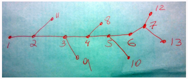
There is nothing we can remove here! Is there something special in the "geometry" of this graph? Yes, it is a tree (a graph in which any two vertices are connected by exactly one path, i.e. connected graph with no circuits - we will talk much more about those guys in the future). Thus a minimal connected set we are looking for should be a tree.

Example: Network's vulnerability

Suppose we are given a graph representing a network of telephone lines. We are interested in networks' vulnerability to accidental distraction we want to identify those lines and switching centers that must stay in service to avoid disconnecting the network.

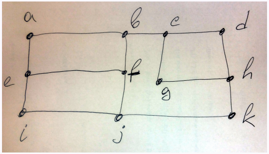


Actually, it is clear that we can remove any edge in this graph and keep the network connected. The same if we remove a vertex (when you remove a vertex, you must remove all edges attached to this vertex). It is also easy to see that we can remove two edges and this will disconnect the network (for example (c, f) and (f, d) will make f to be disconnected from the rest of the vertices). We can ask a different question: what is the minimal number of edges we need to *keep* so that the network is still connected? Lets first play with some graphs:

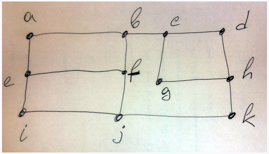


There is nothing we can remove here! Is there something special in the "geometry" of this graph? Yes, it is a tree (a graph in which any two vertices are connected by exactly one path, i.e. connected graph with no circuits - we will talk much more about those guys in the future). Thus a minimal connected set we are looking for should be a tree.

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?

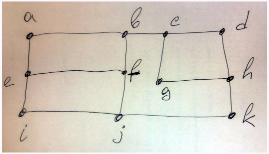


The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



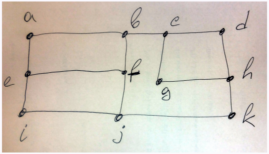
Lets first understand the minimal number of police we may need:

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



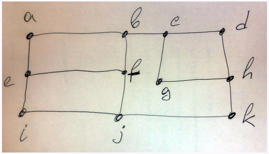
Lets first understand the minimal number of police we may need: The graph has 14 edges.

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each

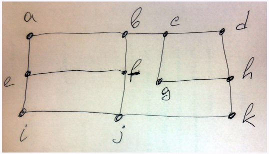
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3).

Police and edge cover

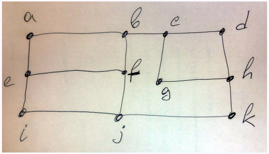
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2).

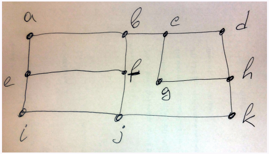
Police and edge cover

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



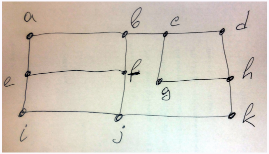
Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



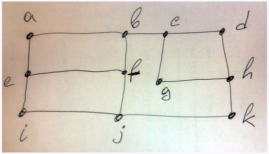
Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$).

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police.

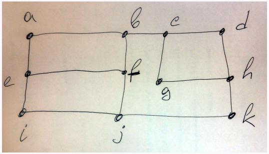
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done.

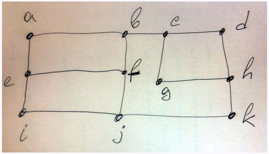
Police and edge cover

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!!

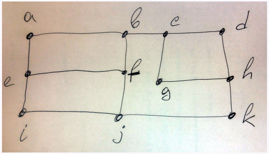
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!! If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices).

Police and edge cover

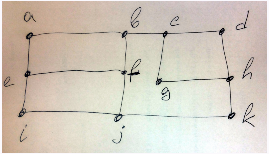
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!! If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once.

Police and edge cover

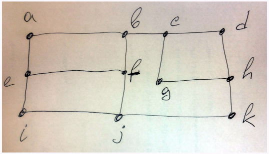
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!! If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job.

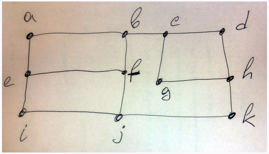
Police and edge cover

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!! If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess).

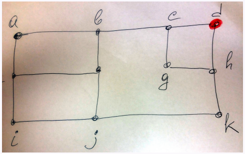
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. Lets try it!! If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)).

Police and edge cover

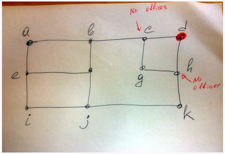
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are ends of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are ends of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d.

Police and edge cover

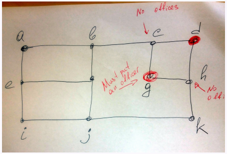
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police).

Police and edge cover

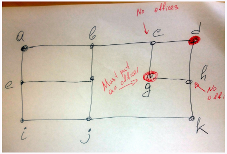
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g.

Police and edge cover

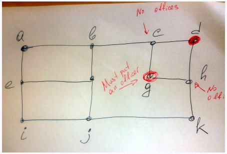
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police).

Police and edge cover

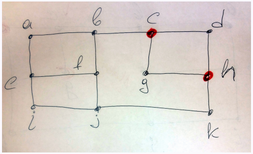
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d.

Police and edge cover

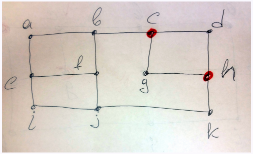
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d. But we MUST take care of edges (c, d) and (d, h).

Police and edge cover

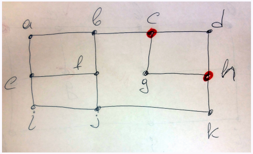
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d. But we MUST take care of edges (c, d) and (d, h). Now look at k we should not put police there! (because, if we do we have and edge with two police and a vertex of degree 2 with a police).

Police and edge cover

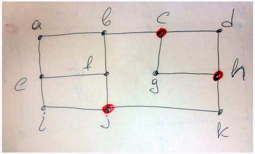
The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d. But we MUST take care of edges (c, d) and (d, h). Now look at k we should not put police there! (because, if we do we have and edge with two police and a vertex of degree 2 with a police). Thus we must put police at j.

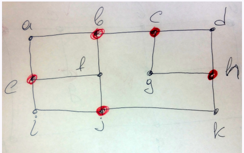
Police and edge cover

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



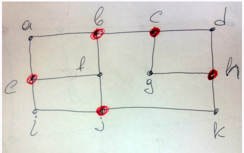
Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d. But we MUST take care of edges (c, d) and (d, h). Now look at k we should not put police there! (because, if we do we have and edge with two police and a vertex of degree 2 with a police). Thus we must put police at j. We finish using the same reasoning.

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



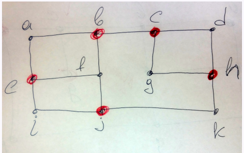
Lets first understand the minimal number of police we may need: The graph has 14 edges. Vertices b, c, e, f, h and h are the end of 3 edges each (i.e. each has degree 3). Vertices a, d, g, i and k are the end of 2 edges each (i.e. each has degree 2). Thus if put police at just 4 vertices they will NOT be able to do the job (indeed police can observe maximum 3 vertices from a given edge, thus 4 police can observe maximum $4 * 3 < 14$). So we need at least 5 police. IF we can find a good position of 5 police then we are done. If we put 5 police all at vertices of degree 3 then we cover $5 * 3 = 15$ edges and thus one edge must be covered twice (i.e. at both vertices). If 4 police at vertices of degree 3 and one of degree 2 then we cover at most 14 edges and must cover every edge exactly once. No other combination will do the job. Now we can start the systematic analysis (i.e. "very" clever guess). Start with an edge containing a vertex of degree 2 (say (c, d)). Assume there is an officer at vertex d. Then we can not put police at c or h (no edge can have both ends with police). Thus we MUST put police to vertex g. CONTRADICTION (we have two vertices of degree 2 with police). Thus we MUST start all over again and there is no police at d. But we MUST take care of edges (c, d) and (d, h). Now look at k we should not put police there! (because, if we do we have and edge with two police and a vertex of degree 2 with a police). Thus we must put police at j. We finish using the same reasoning.

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Police and edge cover (definition)

The graph below represents a part of a city's map. We want to position police at corners (vertices) so that they can keep every block (edge) under surveillance (or mathematically, every edge should have a police at at least one of its vertices). What is the fewest number of police that can do the job?



Edge Cover

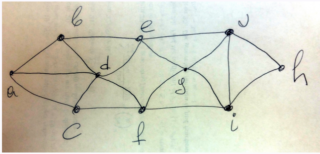
A set C of vertices (i.e. $C \subset V$) in graph G with property that every edge of G is incident to at least one vertex in C is called an edge cover.

In the above graph we see the edge cover of minimal size.

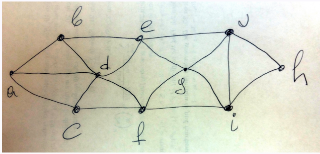
Assume Math. Dep. at Aurora State University has a following schedule problem. As any department they have a lot of committees that meet for one hour each week. One wants to schedule of committee meeting times that minimizes the total number of hours but such that two committees with overlapping members do not meet at the same time.

Committee meetings and independent set of vertices

Assume Math. Dep. at Aurora State University has a following schedule problem. As any department they have a lot of committees that meet for one hour each week. One wants to schedule of committee meeting times that minimizes the total number of hours but such that two committees with overlapping members do not meet at the same time. We will model it with a graph, where each committee is represented by a vertex and joint two vertices by an edge if they represent committees with the same faculty:



Assume Math. Dep. at Aurora State University has a following schedule problem. As any department they have a lot of committees that meet for one hour each week. One wants to schedule of committee meeting times that minimizes the total number of hours but such that two committees with overlapping members do not meet at the same time. We will model it with a graph, where each committee is represented by a vertex and joint two vertices by an edge if they represent committees with the same faculty:

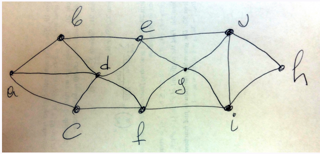


A set of committees can all meet at the same time IF there are no edges between the corresponding set of vertices.

Independent set

A set of vertices without an edge between any two is called an independent set of vertices.

Assume Math. Dep. at Aurora State University has a following schedule problem. As any department they have a lot of committees that meet for one hour each week. One wants to schedule of committee meeting times that minimizes the total number of hours but such that two committees with overlapping members do not meet at the same time. We will model it with a graph, where each committee is represented by a vertex and joint two vertices by an edge if they represent committees with the same faculty:



A set of committees can all meet at the same time IF there are no edges between the corresponding set of vertices.

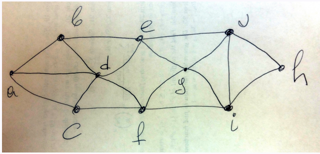
Independent set

A set of vertices without an edge between any two is called an independent set of vertices.

We want to minimize the total number of hours! So we want as many as possible committees meet at the same time. Thus we need to find largest possible independent set(s).

Committee meetings and independent set of vertices

Assume Math. Dep. at Aurora State University has a following schedule problem. As any department they have a lot of committees that meet for one hour each week. One wants to schedule of committee meeting times that minimizes the total number of hours but such that two committees with overlapping members do not meet at the same time. We will model it with a graph, where each committee is represented by a vertex and joint two vertices by an edge if they represent committees with the same faculty:



A set of committees can all meet at the same time IF there are no edges between the corresponding set of vertices.

Independent set

A set of vertices without an edge between any two is called an independent set of vertices.

We want to minimize the total number of hours! So we want as many as possible committees meet at the same time. Thus we need to find largest possible independent set(s). This is far from trivial. If we play a bit with the above example we will see that that there are two independent sets of size 4: a, e, f, h and b, c, g, h and every other independent set will have at most 3 vertices.

Edge Cover and Independent set

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Proof: Note that if there is no edges between the vertices in I, then every edge in E must involve at least one vertex which is not in I (i.e. in $V \setminus I$).

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Proof: Note that if there is no edges between the vertices in I, then every edge in E must involve at least one vertex which is not in I (i.e. in $V \setminus I$). Thus $V \setminus I$ is an edge cover.

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Proof: Note that if there is no edges between the vertices in I, then every edge in E must involve at least one vertex which is not in I (i.e. in $V \setminus I$). Thus $V \setminus I$ is an edge cover.

Now assume $V \setminus I$ is an edge cover.

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Proof: Note that if there is no edges between the vertices in I, then every edge in E must involve at least one vertex which is not in I (i.e. in $V \setminus I$). Thus $V \setminus I$ is an edge cover.

Now assume $V \setminus I$ is an edge cover. Then every edge must have a vertex from $V \setminus I$ and so we can not have an edge with both ends from I, thus no two vertices from I are connected by an edge.

$\square$

A very interesting outcome of this theorem is that if I is an independent set of largest size in G then $V \setminus I$ is an edge cover of smallest possible size.

Theorem

Consider a graph $G = (V, E)$ then $I \subset V$ is an independent set if and only if $V \setminus I$ is an edge cover.

Proof: Note that if there is no edges between the vertices in I, then every edge in E must involve at least one vertex which is not in I (i.e. in $V \setminus I$). Thus $V \setminus I$ is an edge cover.
Now assume $V \setminus I$ is an edge cover. Then every edge must have a vertex from $V \setminus I$ and so we can not have an edge with both ends from I, thus no two vertices from I are connected by an edge.

$\square$

A very interesting outcome of this theorem is that if I is an independent set of largest size in G then $V \setminus I$ is an edge cover of smallest possible size. SO finding maximal independent set is equivalent to finding a minimal edge cover.