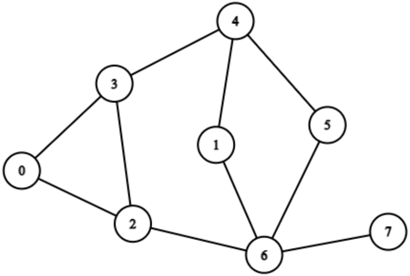
Note on Perfect Forests

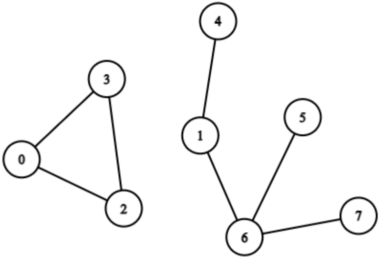
Table of contents

Table of contents

Definition

A spanning subgraph is a subgraph that contains all the vertices of the original graph.

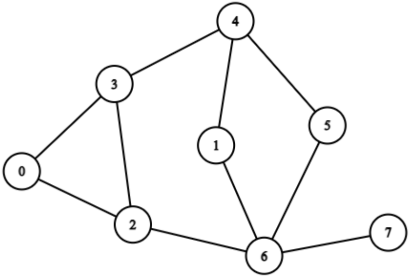


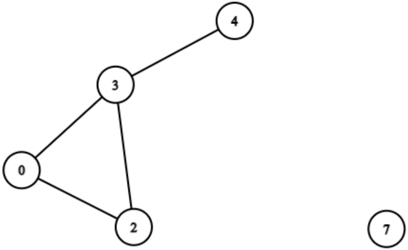


Introduction

Definition

An induced subgraph is a subgraph, formed from a subset of the vertices of the graph and all of the edges (from the original graph) connecting pairs of vertices in that subset.

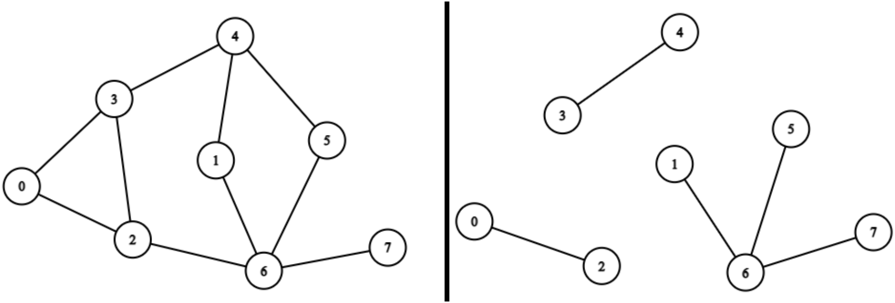




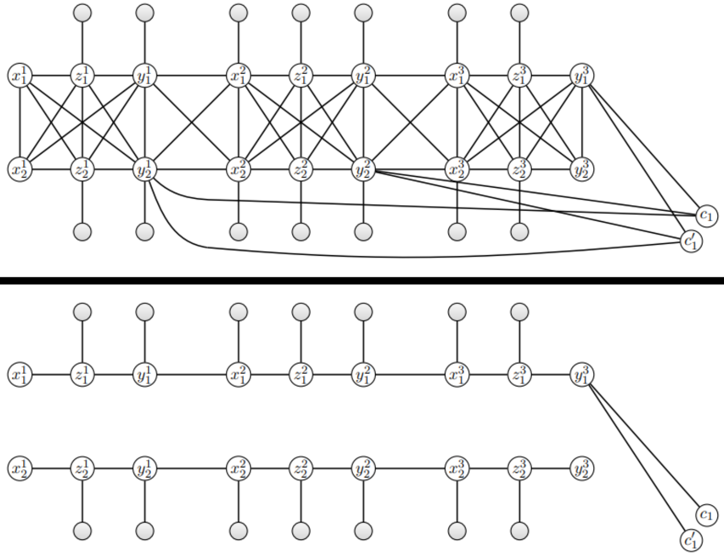
Definition

A spanning subgraph F of a graph G is called a perfect forest if

1. F is a forest
2. the degree $d_F(x)$ of each vertex x in $V(F)$ is odd
3. each tree of F is an induced subgraph of G



Introduction



Introduction

Theorem

A connected graph G contains a perfect forest if and only if G has an even number of vertices.

Table of contents

Proof

It is easy to see that if a connected graph G has a perfect forest, then G has an even number of vertices.

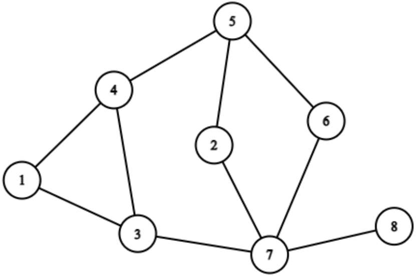
Let's assume that there exist connected graph $G = (V, E)$ that has a perfect forest F and $|V| = 2m + 1$.
By handshaking lemma:

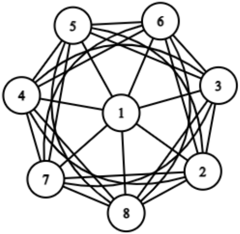
$$\sum_{i=1}^{2m+1} d_F(v_i) = 2|E_F|$$

This contradicts the fact that degree $d_F(x)$ of each vertex x in $V(F)$ is odd.

Proof

Let G be a connected graph of even order n.
Let $V(G) = \{1, ..., n\}$ and let K_n be the complete graph with those n vertices.





Proof

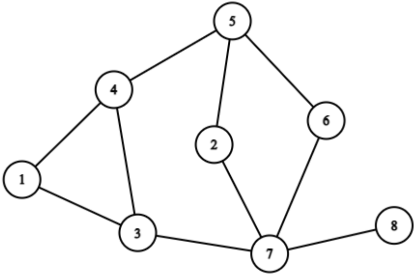
$$\mathbb{F}_2^n = \{0,1\}^n$$

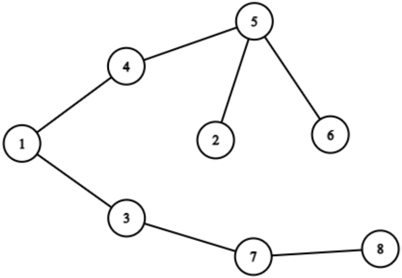
Assuming that $e_{i,j}$ is an edge with vertices i and j, let $v(e_{i,j})$ be a vector in $\mathbb{F}_2^n$ in which the only nonzero coordinates are i and j.

$$v(e_{i,j}) = \left[\overbrace{\underbrace{0,...,0,}_{i-1}1,0,...,0,}^{j-1}1,0,...,0\right]$$

Proof

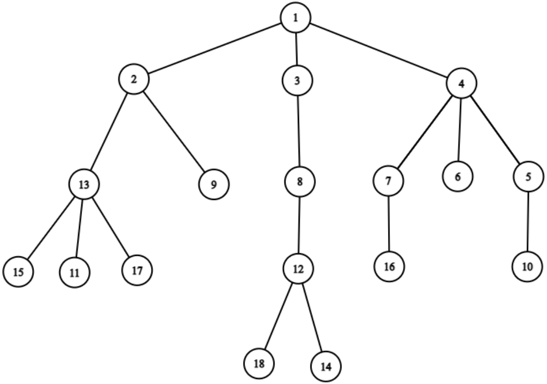
Let T be a spanning tree of G. Vectors $\{v(e) : e \in E(T)\}$ are linearly independent.





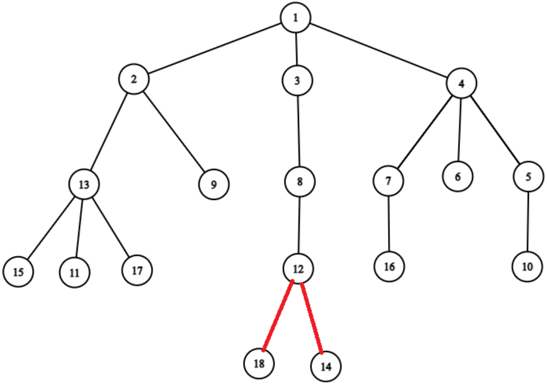
Proof

To show independency we need to show that there does not exist subset of $\{v(e) : e \in E(T)\}$ in which the sum of the vectors is zero. It can be done by analyzing the rooted tree T from the bottom up. In the first step, we conclude that this subset cannot contain edges with leaves and so on.



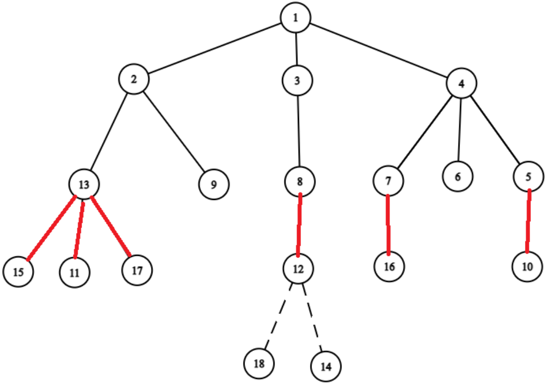
Proof

To show independency we need to show that there does not exist subset of $\{v(e) : e \in E(T)\}$ in which the sum of the vectors is zero. It can be done by analyzing the rooted tree T from the bottom up. In the first step, we conclude that this subset cannot contain edges with leaves and so on.



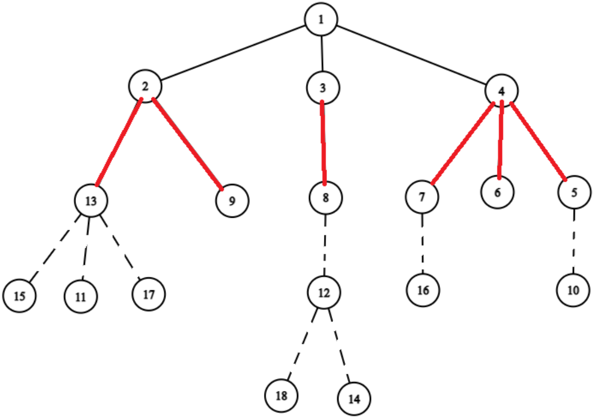
Proof

To show independency we need to show that there does not exist subset of $\{v(e) : e \in E(T)\}$ in which the sum of the vectors is zero. It can be done by analyzing the rooted tree T from the bottom up. In the first step, we conclude that this subset cannot contain edges with leaves and so on.



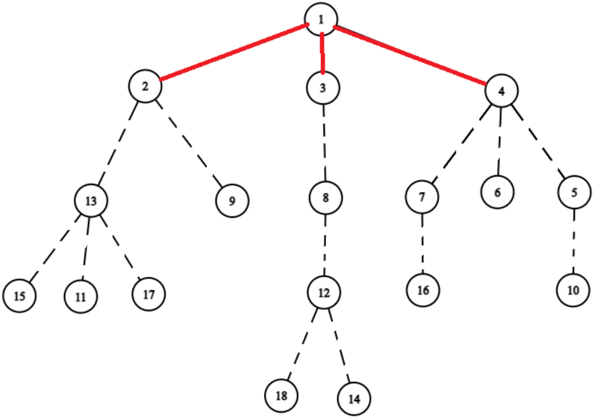
Proof

To show independency we need to show that there does not exist subset of $\{v(e) : e \in E(T)\}$ in which the sum of the vectors is zero. It can be done by analyzing the rooted tree T from the bottom up. In the first step, we conclude that this subset cannot contain edges with leaves and so on.



Proof

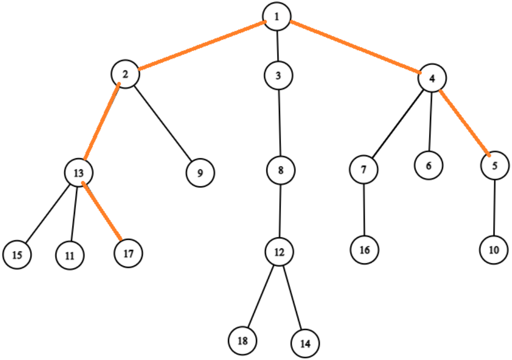
To show independency we need to show that there does not exist subset of $\{v(e) : e \in E(T)\}$ in which the sum of the vectors is zero. It can be done by analyzing the rooted tree T from the bottom up. In the first step, we conclude that this subset cannot contain edges with leaves and so on.



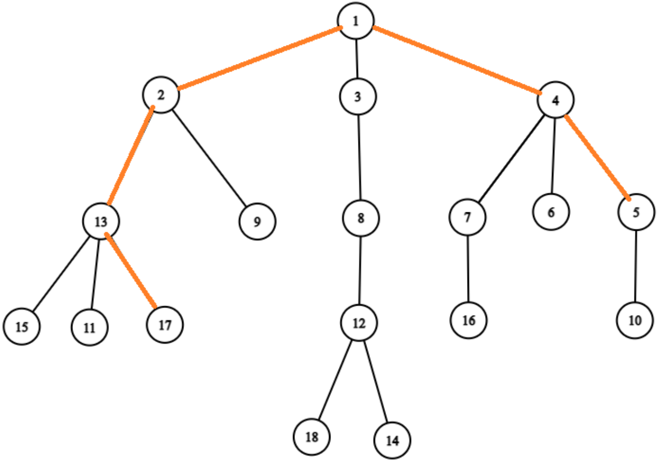
Proof

Another observation is that $\forall e \in E(K_n) : v(e)$ may be written as a linear combination of vectors in $\{v(e) : e \in E(T)\}$.
To do that for some $e_{i,j}$ we can add all the vectors that correspond to the edges in the path between i and j.

$$v(e_{i,j}) + v(e_{j,k}) = v(e_{i,k})$$



Proof



$$v(e_{5,17}) = v(e_{5,4}) + v(e_{4,1}) + v(e_{1,2}) + v(e_{2,13}) + v(e_{13,17})$$

Proof

By using previous observation we know that vectors $v(e_{1,2}), v(e_{3,4}), ..., v(e_{n-1,n})$ are linear combinations of vectors in $\{v(e) : e \in E(T)\}$.

$v(e_{1,2}) + v(e_{3,4}) + ... + v(e_{n-1,n}) = [1, ..., 1]$.
Thus, there exists $L \subseteq \{v(e) : e \in E(T)\}$ such that:

$$\sum_{v \in L} v = [1, ..., 1]$$

We know that vectors in $\{v(e) : e \in E(T)\}$ are linearly independent, so vectors in L must also be linearly independent.

Proof

Let $M = \{v(e) : e \in E(G)\}$, and for each $v(e) \in M \setminus L$, let
$S_{v(e)} = v(e) \cup L$.
We will consider two cases:

1° $S_{v(e)}$ is linearly dependent for some $v(e) \in M$.
Then we can find $L' \subseteq L$ such that $|L'| > 1$ and $v(e) = \sum_{v \in L'} v$.

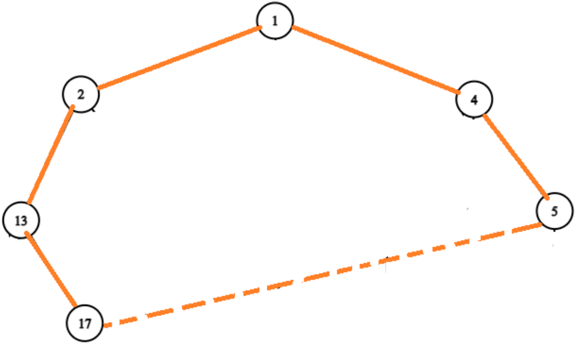
In that case let $L := \{v(e)\} \cup (L \setminus L')$. The new L has fewer elements, but still $\sum_{v \in L} v = [1, ..., 1]$ and vectors in L are linearly independent.

Proof

2° $S_{v(e)}$ is linearly independent for each $v(e) \in M \setminus L$.
In order to construct perfect forest of graph G we can take subgraph F induced by the edges $\{e \in E(G) : v(e) \in L\}$.

Proof

Sum of vectors corresponding to the edges of a cycle is zero, but L is linearly independent, so F doesn't have cycles $\Rightarrow$ F is forest.



Proof

$\sum_{v \in L} v = [1, \ldots, 1] \Rightarrow$ the degree $d_F(x)$ of each vertex x in $V(F)$ is odd.

Proof

If i and j are in the same tree in F and $e_{i,j} \notin E(F)$ then $e_{i,j} \notin E(G)$. That's because $S_{v(e)}$ is linearly independent for each $v(e) \in M \setminus L$, so $v(e_{i,j}) \notin M \setminus L \Rightarrow$ each tree of F is an induced subgraph of G.

Proof

Since 1° produces smaller L and $|L| \leqslant n$, after at most n iterations of 1° we will arrive at 2°.

$\blacksquare$

Table of contents

Corollary

This proof can be turned into a polynomial algorithm to find a perfect forest as there are polynomial algorithms to check linear independence of vectors and, if the vectors are linearly dependent, to find a nontrivial linear combination of them which is equal to the zero vector.

Corollary

Note that a perfect matching is a perfect forest. A perfect forest can be thought of as a generalization of a perfect matching since, in a matching, all components are trees on one edge.

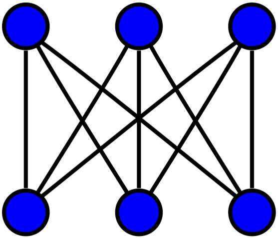
Table of contents

Other properties

Theorem

Perfect matching problem for bipartite cubic graphs belongs to the complexity class NC.

Sharan and Wigderson used structure of perfect forests (they called it pseudo-matching) to prove that.
NC can be thought of as the problems that can be efficiently solved on a parallel computer.



Other properties

Theorem

In polynomial time, we can find a perfect forest of minimum size.

Theorem

It is NP-hard to find a perfect forest of maximum size.

Other properties

Theorem

The problem of finding a perfect forest of size at least $n-1$ is polynomial-time solvable.

Theorem

It is NP-hard to decide whether a connected graph contains a perfect forest with at least $n-2$ edges.

Gregory Gutin and Anders Yeo proved it using reduction from NAE-3-SAT. It is easy to show that this Theorem holds if we replace $n-2$ by $n-k$ for any integer $k \geqslant 2$.

Other properties

Theorem

Given a graph G and an edge $e \in E(G)$ we can in polynomial time decide whether G has a perfect forest not containing e.

Theorem

The following problem is NP-hard. Given a connected graph G and an edge $e \in E(G)$, decide whether G has a perfect forest containing e.

Table of contents

Extension

Definition

For $i \in \{0, 1\}$ and a connected graph G, a spanning forest F of G is called an i-perfect forest if every tree in F is an induced subgraph of G and exactly i vertices of F have even degree (including zero). A i-perfect forest of G is proper if it has no vertices of degree zero.

So 0-perfect forest is defined exactly the same as perfect forest. 1-perfect forest is almost the same but it contains exactly 1 vertex with even degree.

Table of contents

References

Gregory Gutin - Note on Perfect Forests

Roded Sharan, Avi Wigderson - A new NC Algorithm for Perfect Matching in Bipartite Cubic Graphs

Gregory Gutin, Anders Yeo - Perfect Forests in Graphs and Their Extensions

The End

Thank you!