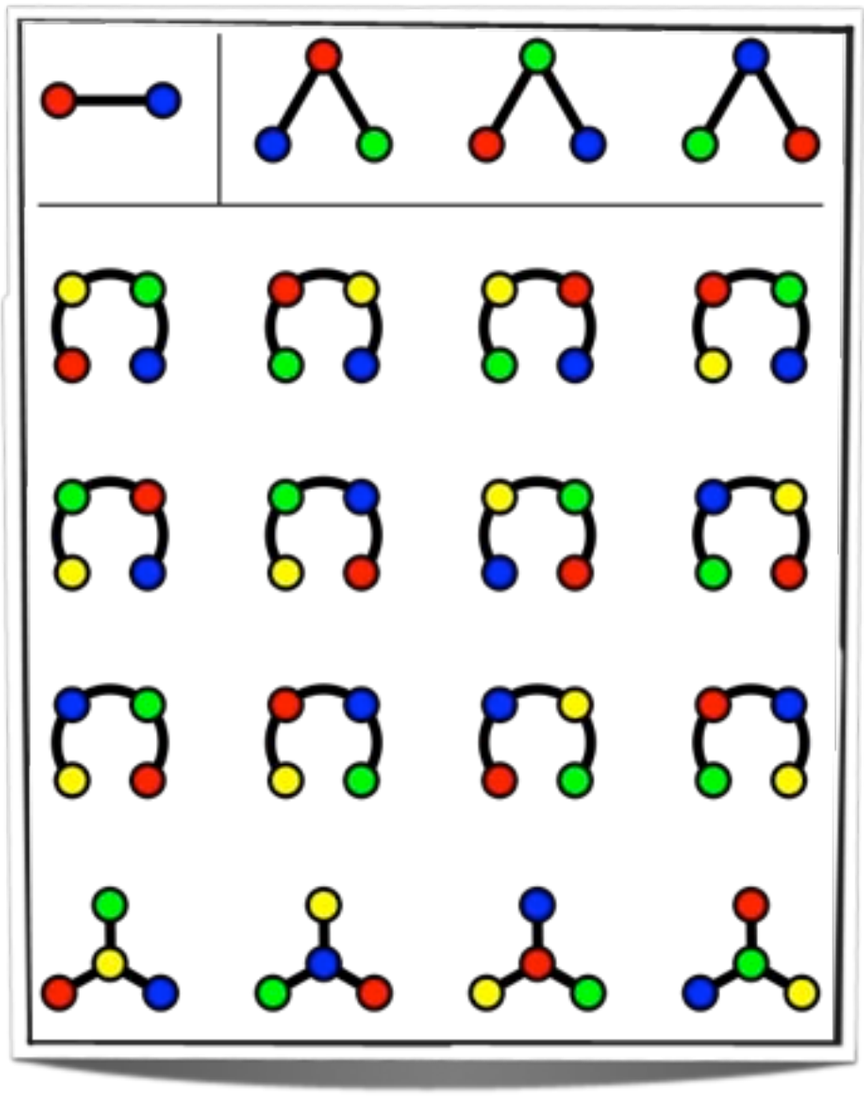
Combinatorics

南京大学
尹一通

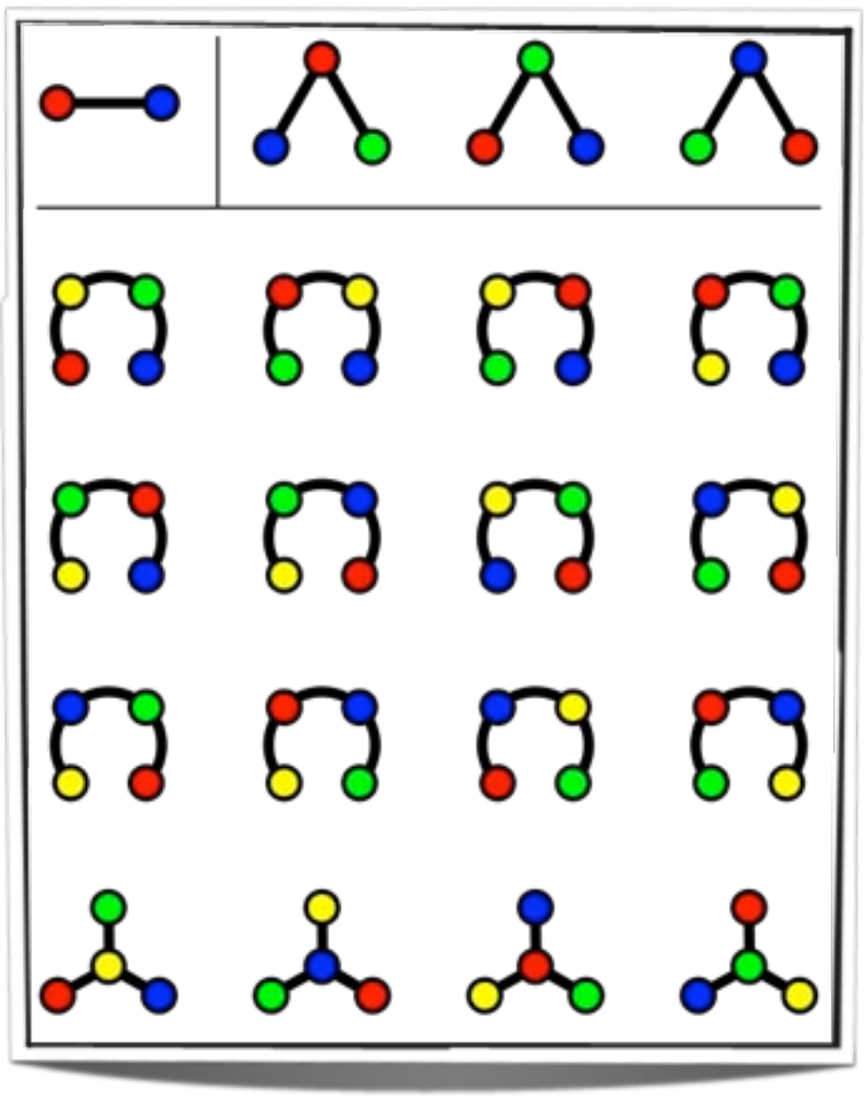
Counting (labeled) trees

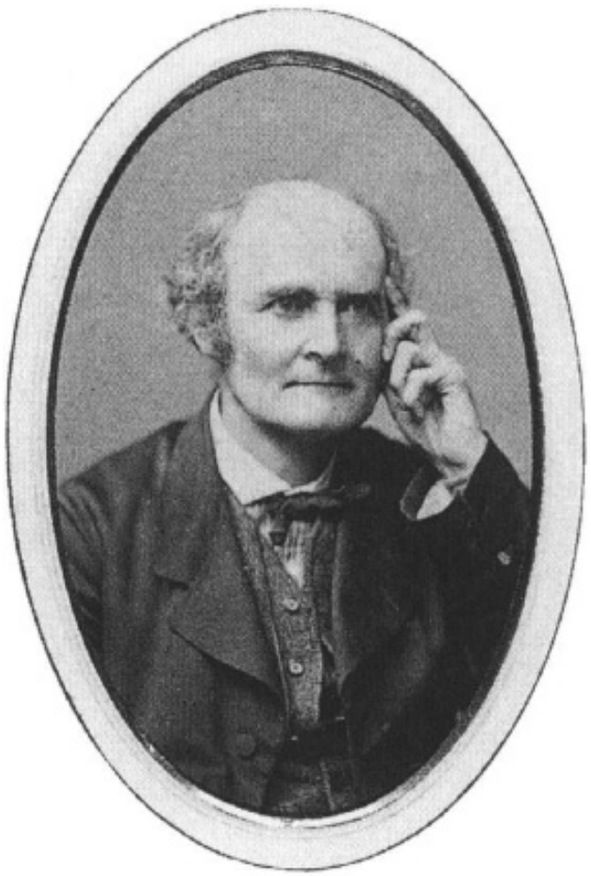


"How many different trees can be formed from n distinct vertices?"

Cayley's formula:

There are n^{n-2} trees on n distinct vertices.





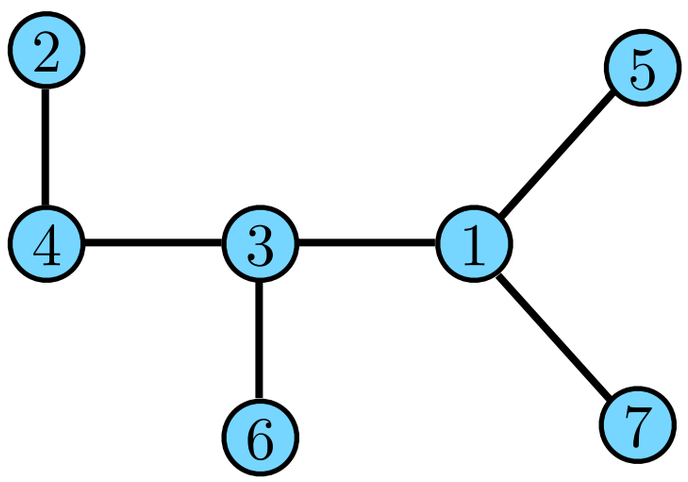
Arthur Cayley

Prüfer Code

leaf : vertex of degree 1

removing a leaf from T, still a tree

T_1 :



$T_1 = T$;

for $i = 1$ to n-1

u_i : smallest leaf in T_i ;

(u_i, v_i) : edge in T_i ;

T_{i+1} = delete u_i from T_i ;

u_i : $2,\ 4,\ 5,\ 6,\ 3,\ 1$

v_i : $4,\ 3,\ 1,\ 3,\ 1,\ 7$

Prüfer code:

$(v_1,\ v_2,\ \ldots\ ,\ v_{n\text{-}2})$

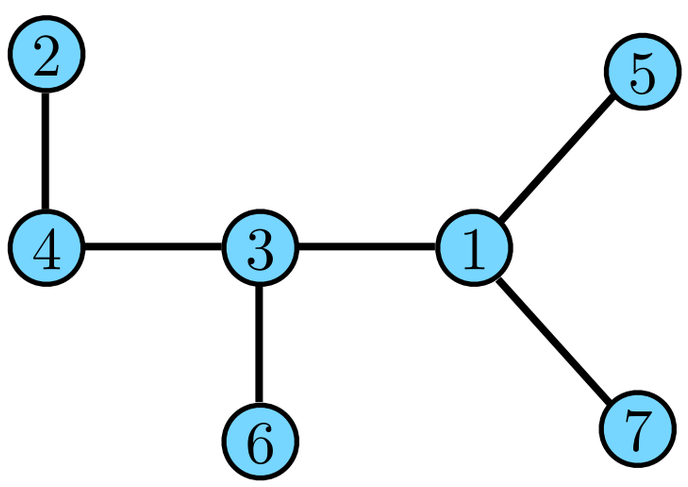
edges of T: (u_i, v_i), $1 \le i \le n\text{-}1$

$v_{n\text{-}1} = n$

u_i: smallest leaf in T_i
a tree has ≥ 2 leaves

$\Rightarrow$ $u_i \neq n$
n is never deleted

T :



u_i: 2, 4, 5, 6, 3, 1

v_i: 4, 3, 1, 3, 1, 7

$(v_1, v_2, \ldots, v_{n\text{-}2})$

Only need to recover every u_i from $(v_1, v_1, \ldots, v_{n\text{-}2})$.

u_i is the smallest number not in $\{u_1, \ldots, u_{i-1}\} \cup \{v_i, \ldots, v_{n-1}\}$

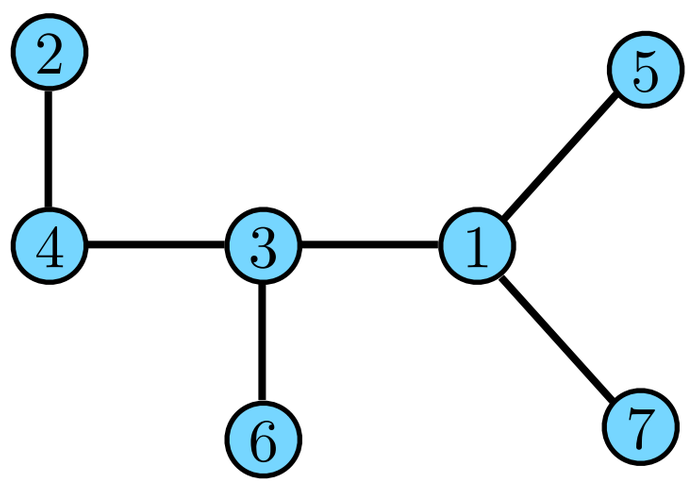
u_i is the smallest number not in $\{u_1, \ldots, u_{i-1}\} \cup \{v_i, \ldots, v_{n-1}\}$

$\forall$ vertex v in T,

occurrences of v in $u_1,\ u_2,\ \ldots\ ,\ u_{n\text{-}1},\ v_{n\text{-}1}$: 1

occurrences of v in edges (u_i, v_i), $1 \le i \le n\text{-}1$: $\deg_T(v)$

T :



occurrences of v in Prüfer code: $(v_1,\ v_2,\ \ldots\ ,\ v_{n\text{-}2})$

$\deg_T(v)\text{-}1$

u_i : $2,\ 4,\ 5,\ 6,\ 3,\ 1$

v_i : $4,\ 3,\ 1,\ 3,\ 1,\ 7$

$(v_1,\ v_2,\ \ldots\ ,\ v_{n\text{-}2})$

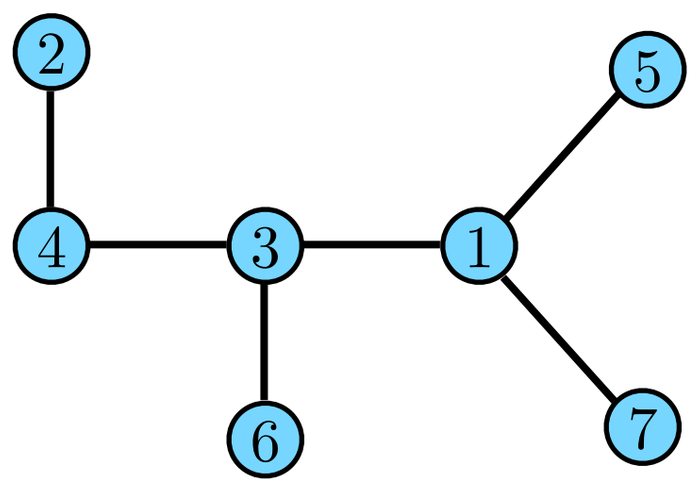
u_i is the smallest number not in $\{u_1, \ldots, u_{i-1}\} \cup \{v_i, \ldots, v_{n-1}\}$

$\forall$ vertex v in T_i,

occurrences of v in $u_i,\ u_{i+1},\ \ldots\ ,\ u_{n\text{-}1},\ v_{n\text{-}1}$: 1

occurrences of v in edges (u_j, v_j), $i \leq j \leq n\text{-}1$: $\deg_{T_i}(v)$

T :



occurrences of v in $(v_i,\ \ldots\ ,\ v_{n\text{-}2})$

$\deg_{T_i}(v) - 1$

u_i : $2,\ 4,\ 5,\ 6,\ 3,\ 1$

v_i : $4,\ 3,\ 1,\ 3,\ 1,\ 7$

$(v_1,\ v_2,\ \ldots\ ,\ v_{n\text{-}2})$

leaf v of T_i :

in $\{u_i,\ u_{i+1},\ \ldots\ ,\ u_{n\text{-}1},\ v_{n\text{-}1}\}$

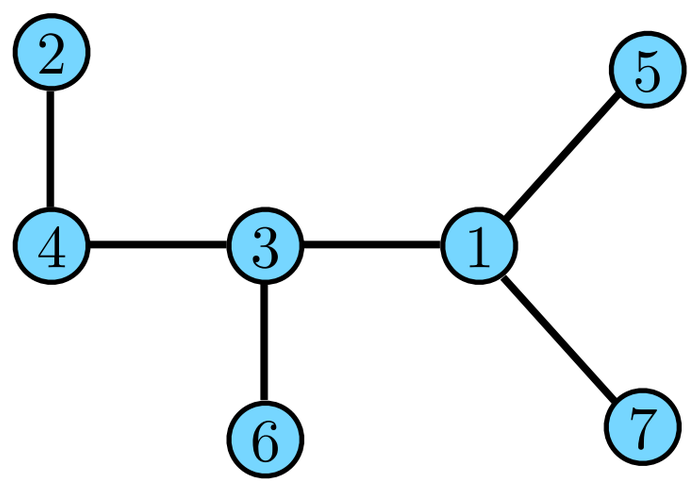
not in $\{v_i,\ v_{i+1},\ \ldots\ ,\ v_{n\text{-}2}\}$

u_i : smallest leaf in T_i

u_i is the smallest number not in

$\{u_1, \ldots, u_{i-1}\} \cup \{v_i, \ldots, v_{n-1}\}$

T :



u_i : 2, 4, 5, 6, 3, 1

v_i : 4, 3, 1, 3, 1, 7

$(v_1,\ v_2,\ \ldots\ ,\ v_{n\text{-}2})$

$T =$ empty graph;

$v_{n\text{-}1} = n$;

for $i = 1$ to n-1

u_i : smallest number not in

$\{u_1,\ldots,u_{i\text{-}1}\} \cup \{v_i,\ldots,v_{n\text{-}1}\}$

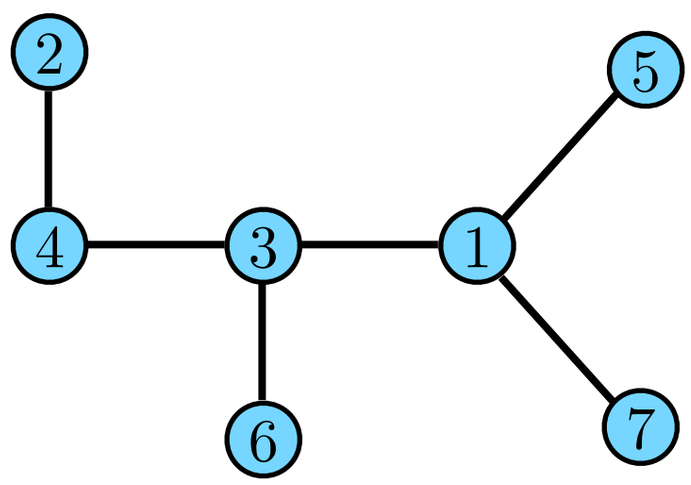
add edge (u_i, v_i) to T;

Prüfer code is reversible ⇒ 1-1

every $(v_1, v_2, \ldots, v_{n-2}) \in \{1, 2, \ldots, n\}^{n-2}$

is decodable to a tree ⇒ onto

T :



u_i : 2, 4, 5, 6, 3, 1

v_i : 4, 3, 1, 3, 1, 7

$(v_1, v_2, \ldots, v_{n-2})$

$T =$ empty graph;

$v_{n-1} = n$;

for $i = 1$ to n-1

u_i : smallest number not in $\{u_1, \ldots, u_{i-1}\} \cup \{v_i, \ldots, v_{n-1}\}$

add edge (u_i, v_i) to T;

Prüfer code is reversible $\Rightarrow$ 1-1

every $(v_1, v_2, \ldots, v_{n-2}) \in \{1, 2, \ldots, n\}^{n-2}$

is decodable to a tree $\Rightarrow$ onto

Cayley's formula:

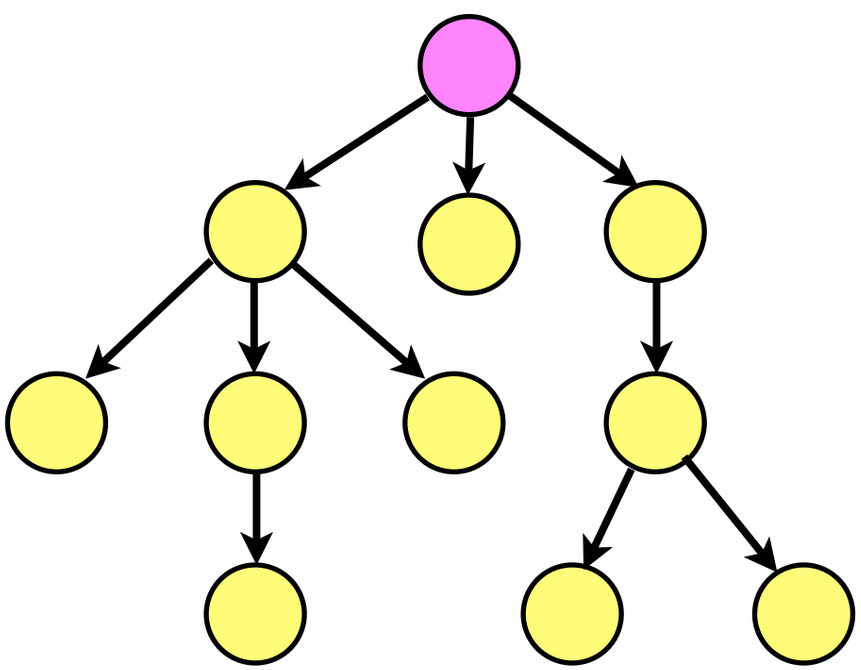
There are n^{n-2} trees on n distinct vertices.

Double Counting

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

T_n : # of trees on *n* distinct vertices.

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*



From a tree:

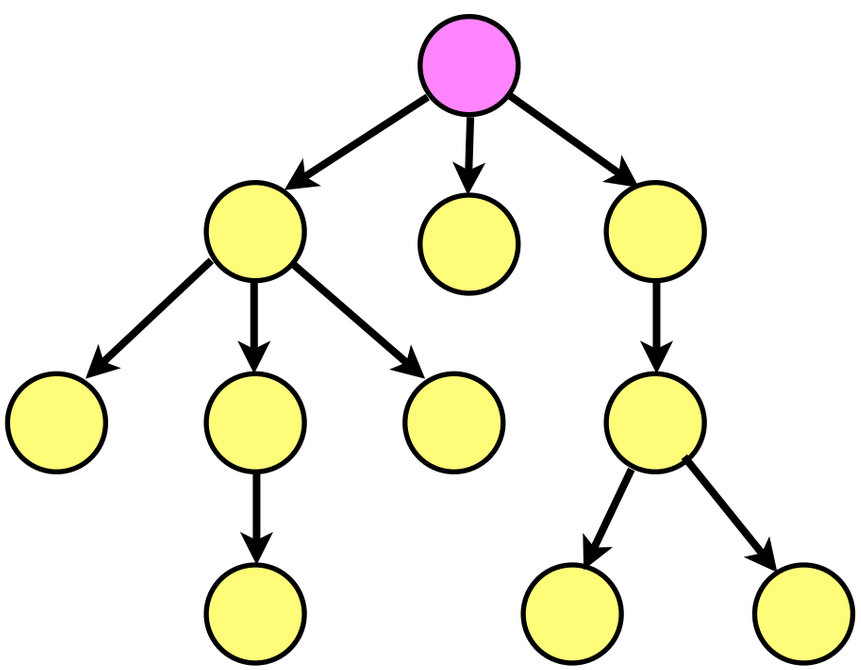
- pick a root;
- pick an order of edges.

$$T_n n(n-1)!$$

$$= n! T_n$$

T_n : # of trees on *n* distinct vertices.

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

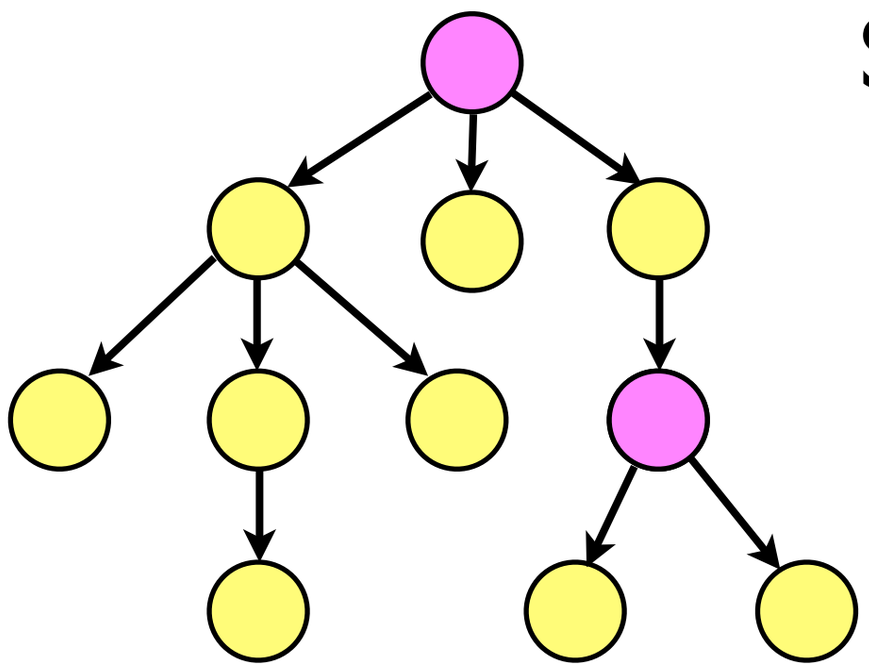


From an empty graph:

- add edges one by one

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

From an empty graph: • add edges one by one

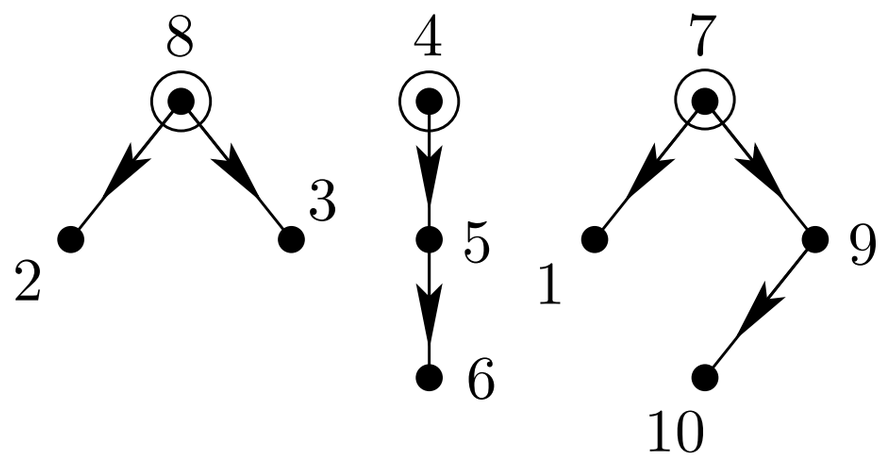


Start from *n* ~~isolated vertices~~ rooted trees

Each step joins 2 trees.

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

From an empty graph: • add edges one by one

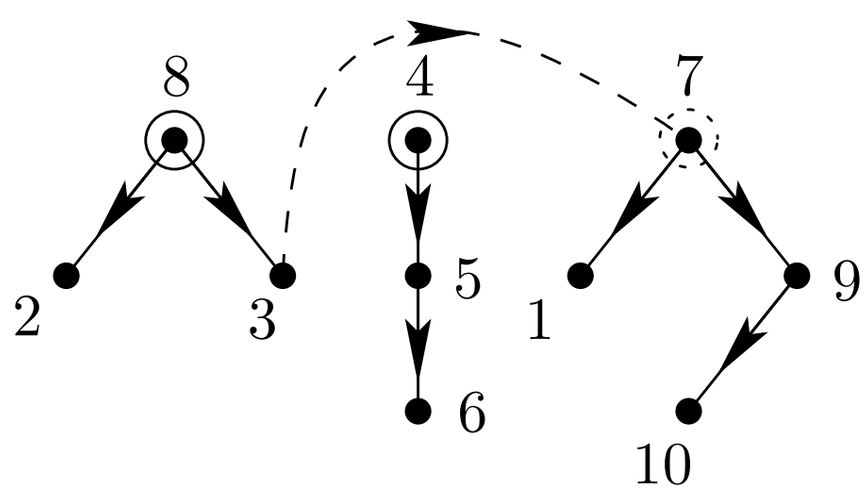


Start from *n* rooted trees.

After adding *n*-*k* edges

k rooted trees

add an edge



any vertex → root of another tree

n *k*-1

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

From an empty graph: • add edges one by one

$$\prod_{k=2}^{n} n(k-1)$$

$$= n^{n-2} n!$$

Start from *n* rooted trees.

After adding *n*-*k* edges

k rooted trees

add an edge

any vertex → root of another tree

n *k*-1

of *sequences* of adding *directed edges* to an empty graph to form a *rooted tree*

From a tree:

- pick a root;
- pick an order of edges.

$$T_n n(n-1)!$$
$$= n! T_n$$

=

From an empty graph:

- add edges one by one

$$\prod_{k=2}^{n} n(k-1)$$
$$= n^{n-2} n!$$

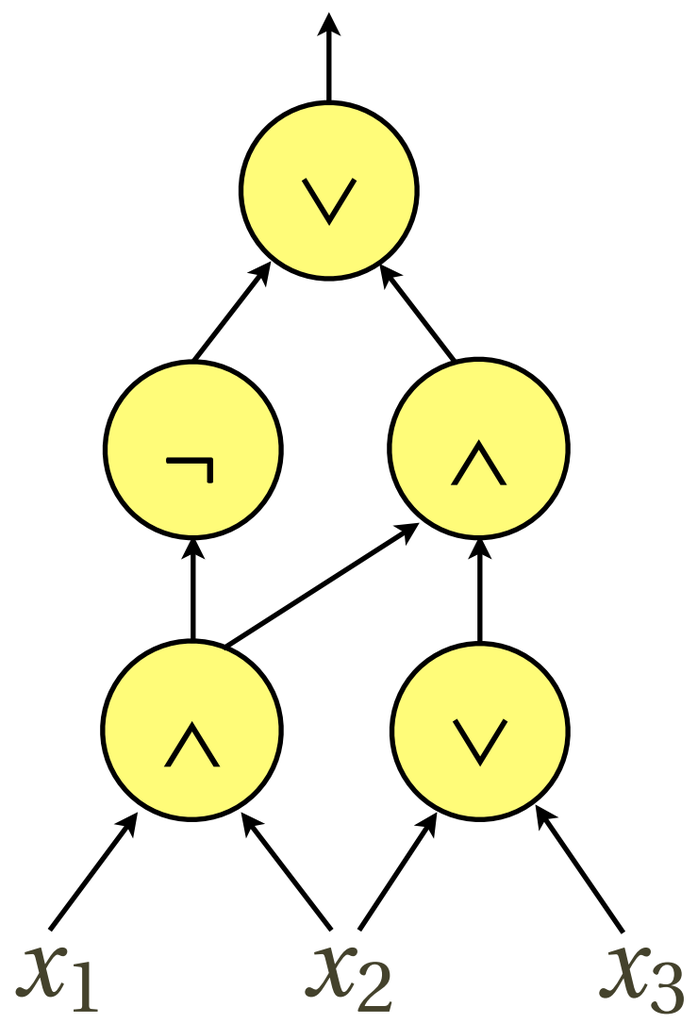
$$T_n = n^{n-2}$$

Existence

Circuit Complexity

Boolean function $f : \{0,1\}^n \to \{0,1\}$

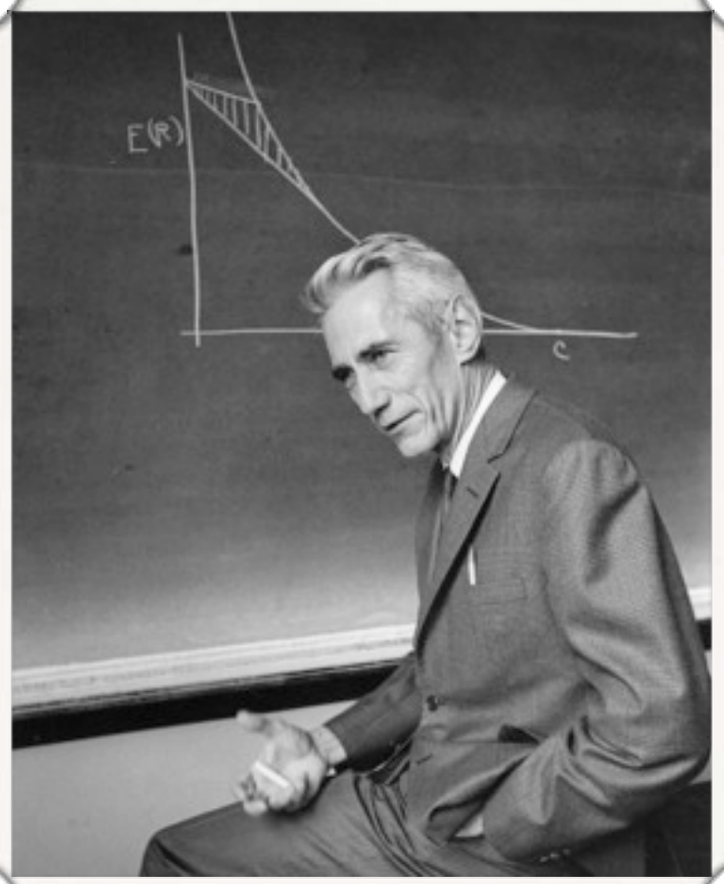
Boolean circuit



- DAG (directed acyclic graph)
- Nodes:
  - inputs: $x_1 \ldots x_n$
  - gates: ∧ ∨ ¬
- Complexity: #gates

Theorem (Shannon 1949)

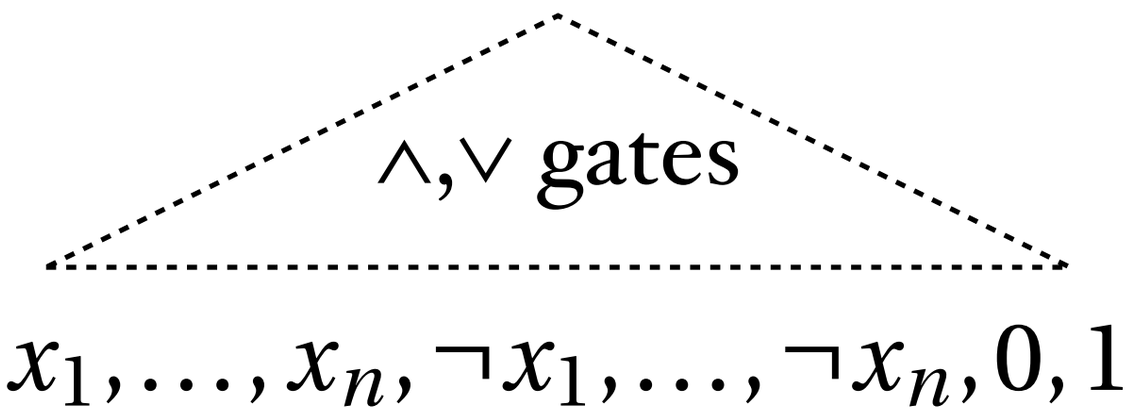
There is a boolean function $f : \{0,1\}^n \to \{0,1\}$ which cannot be computed by any circuit with $\frac{2^n}{3n}$ gates.

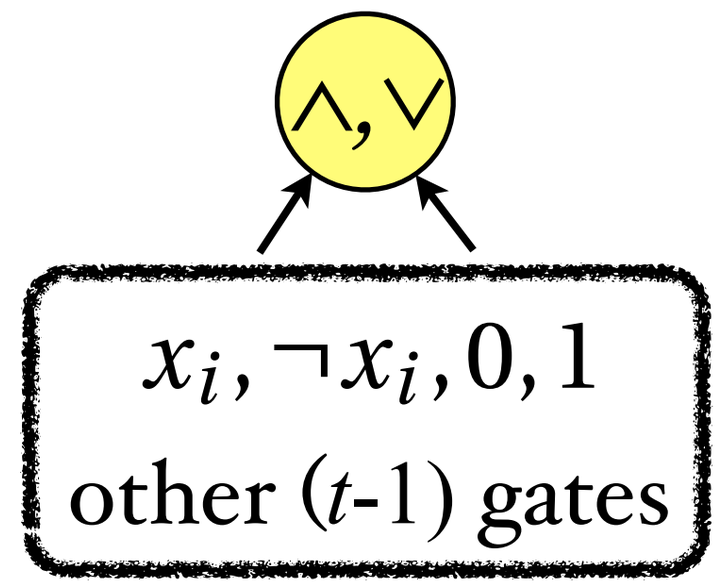


Claude Shannon

of $f : \{0,1\}^n \to \{0,1\}$ $\qquad \left|\{0,1\}^{2^n}\right| = 2^{2^n}$

of circuits with t gates: $\quad < \quad 2^t(2n+t+1)^{2t}$





De Morgan's law:

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

Theorem (Shannon 1949)

Almost all

~~There is a~~ boolean function $f : \{0,1\}^n \to \{0,1\}$ which cannot be computed by any circuit with $\frac{2^n}{3n}$ gates.

one circuit computes one function

$\#f$ computable by t gates $\leq$

$\#$circuits with t gates $\leq$

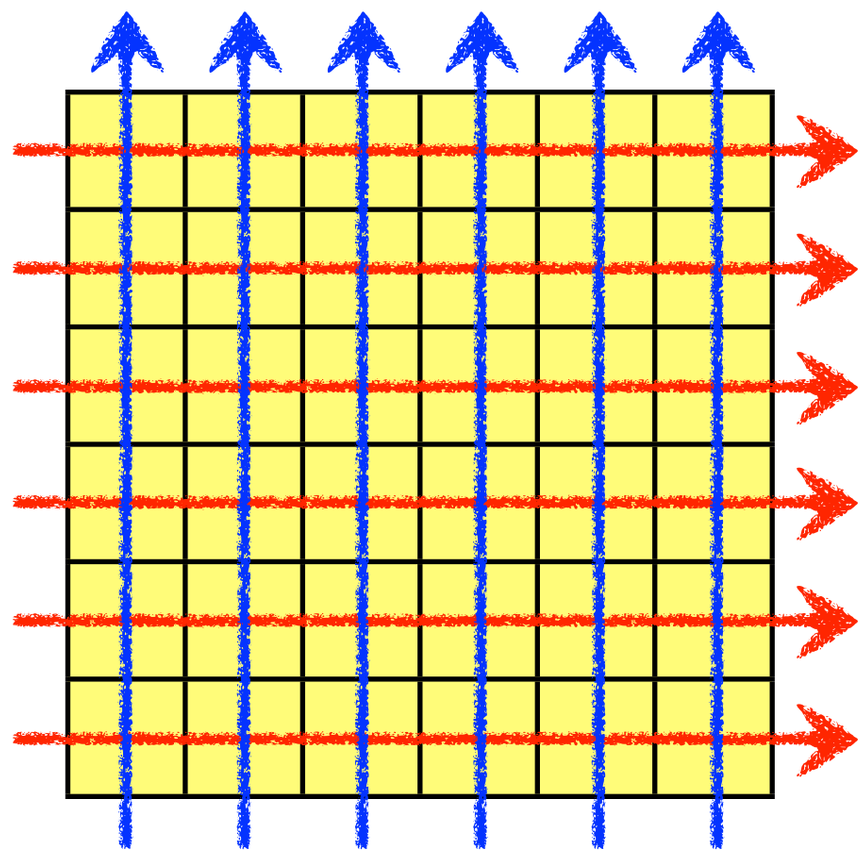
$$2^t(2n+t+1)^{2t} \ll 2^{2^n} = \#f$$

$$t = 2^n/3n$$

Double Counting

"Count the same thing twice.
The result will be the same."

sum by row
A



A=B

sum by column
B

Handshaking lemma

A party of *n* guests.

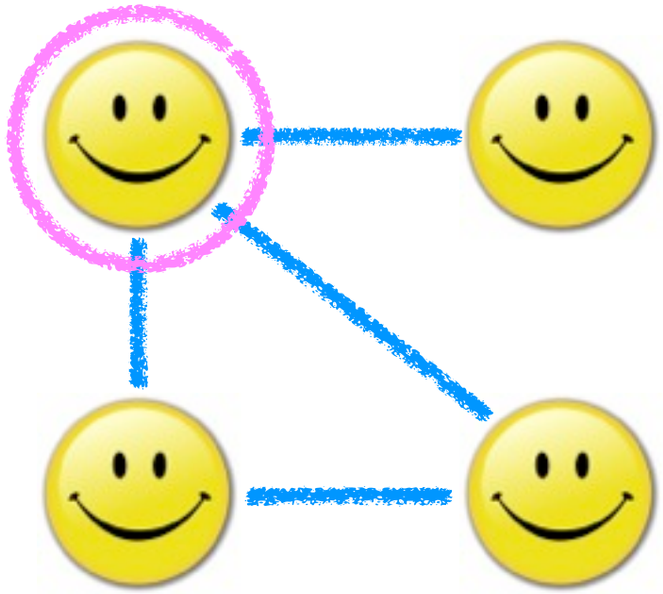
The number of guests who shake hands an odd number of times is even.

Modeling:

n guests ⇔ *n* vertices

handshaking ⇔ edge

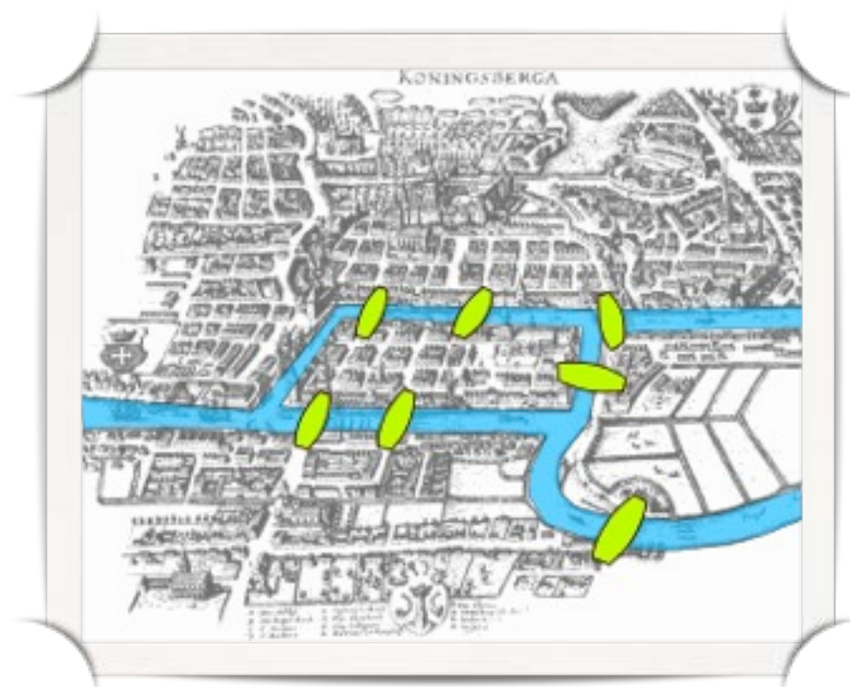
# of handshaking ⇔ degree



Lemma (Euler 1736)

$$\sum_{v \in V} d(v) = 2|E|$$

Leonhard Euler



In the 1736 paper of

Seven Bridges of Königsberg

Lemma (Euler 1736)

$$\sum_{v \in V} d(v) = 2|E|$$

Count directed edges:

$$(u, v) : \{u, v\} \in E$$

Count by vertex:

$\forall v \in V$

d directed edges

$(v, u_1) \cdots (v, u_d)$

=

Count by edge:

$\forall \{u, v\} \in E$

2 directions

(u, v) and (v, u)

Lemma (Euler 1736)

$$\sum_{v \in V} d(v) = 2|E|$$

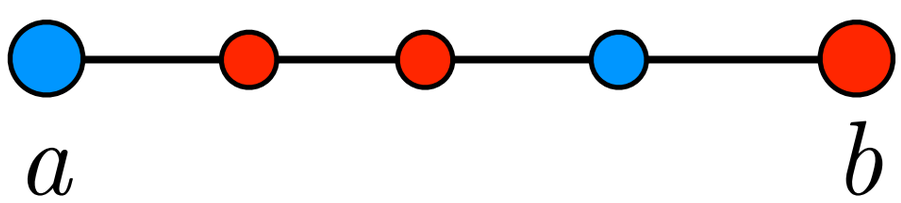
Corollary

# of odd-degree vertices is even.

Sperner's Lemma

line segment: ab divided into small segments

each endpoint: red or blue

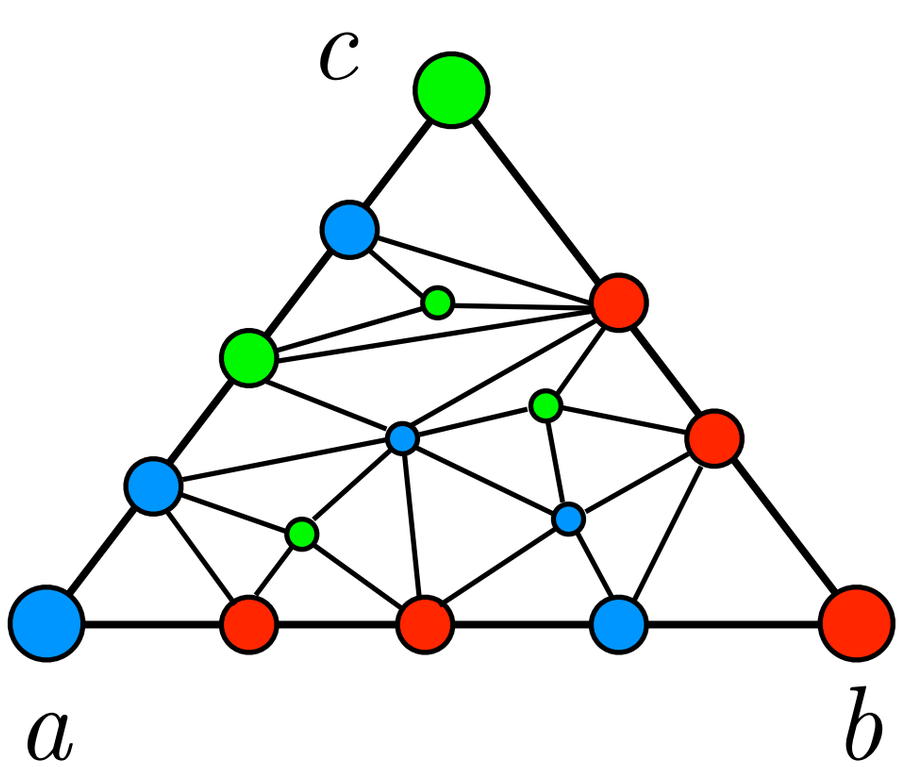


ab have different color

$\exists$ small segment

Emanuel Sperner

Sperner's Lemma



triangle: abc

triangulation

proper coloring:

3 colors red, blue, green

abc is tricolored

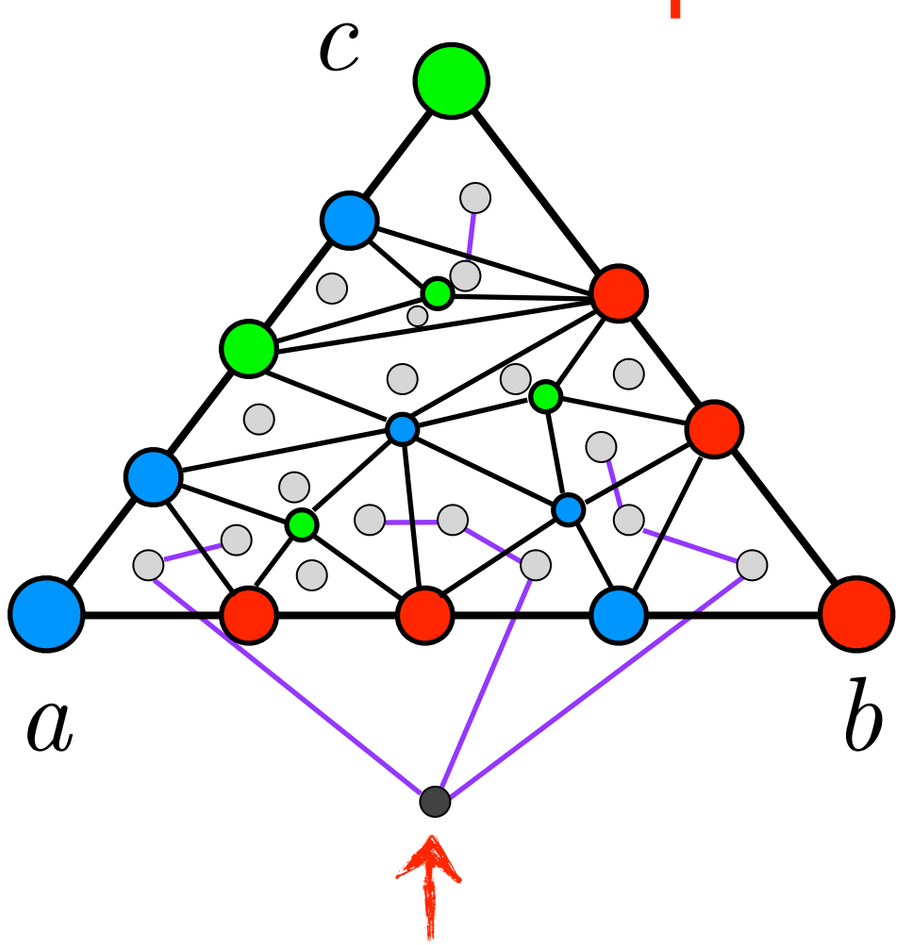
lines ab, bc, ac are 2-colored

Sperner's Lemma (1928)

$\forall$ properly colored triangulation of a triangle,
$\exists$ a tricolored small triangle.

Sperner's Lemma (1928)

∀ properly colored triangulation of a triangle,
∃ a tricolored small triangle.



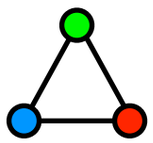
partial dual graph:

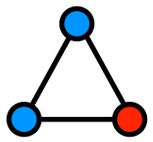
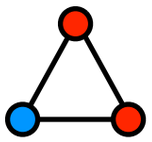
each △ is a vertex

the outer-space is a vertex

add an edge if 2 △

share a ●—● edge

degree of  node: 1

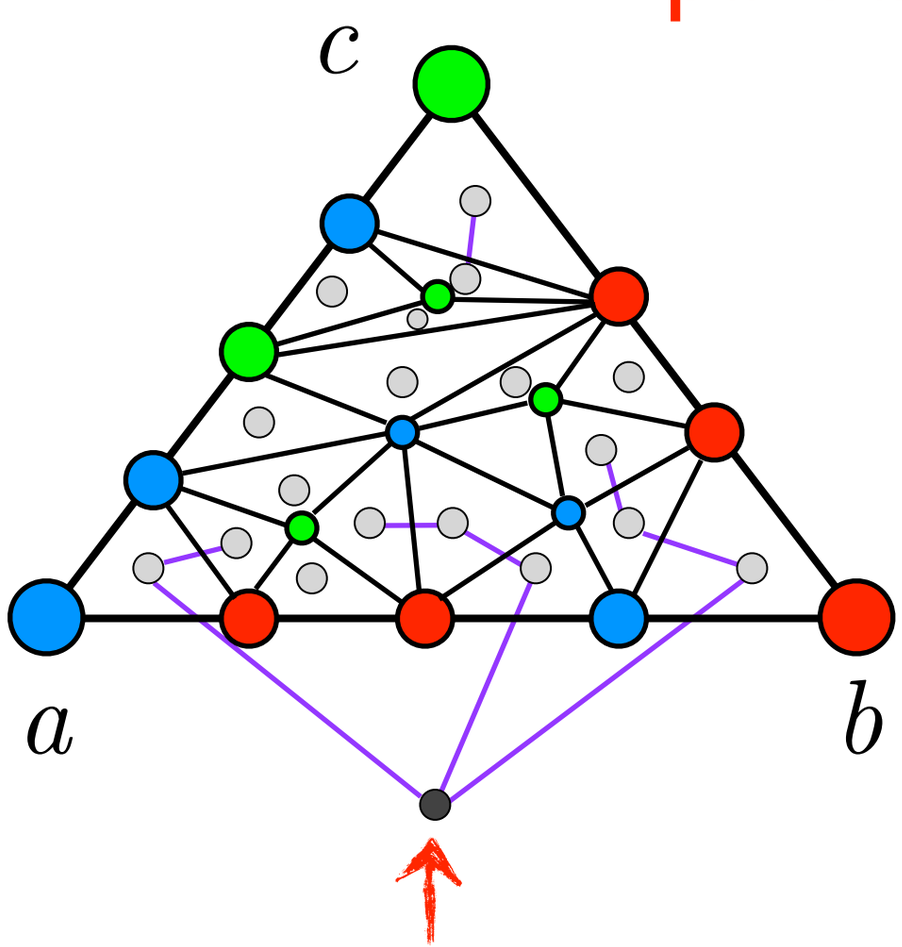
degree of  or  node: 2

other cases: 0 degree

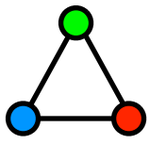
Sperner's Lemma (1928)

∀ properly colored triangulation of a triangle,
∃ a tricolored small triangle.

partial dual graph:



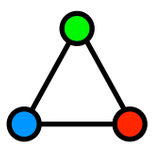
degree is odd

degree of  node: 1

degree of other △: even

handshaking lemma:

# of odd-degree vertices is even.

of  : $\text{odd} \neq 0$

Sperner's Lemma (1928)

∀ properly colored triangulation of a triangle,
∃ a tricolored small triangle.

high-dimension: triangle ⇒ simplex

triangulation ⇒ simplicial subdivision

Brouwer's fixed point theorem (1911)

∀ continuous function f: $B \to B$ of an n-dimensional ball B, ∃ a fixed point $x = f(x)$.