31.1 The handshaking lemma

→ 2 hands form a handshake

→ sum of degree = 2 × num of edges.
[in all vertex]

→ n vertices = (n−1) edges

→ sum of degree must be even.

eg: sum of degree = 10

→ must have 5 edges no matter how it's drawn use all edge

Trail	no repeated edge
Euler Trail	Trail + All edges are used only once (start & end at diff point)
Closed Euler Trail	Euler Trail + Start & end at same vertex

only 2 odd, the rest will be even. 4 cannot

[all vertex must have even degree]

2 vertex with odd degree : can have Euler Trail
start & end at diff pt

31.5 The converse theorem

p → q
premix consequent

closed Euler Trail
→ all vertex are even degree.

[A connected graph with no odd degree vertices has a closed Euler trail.]

31.3 Euler's Solution

→ not possible, because

1. each time a walk enter & come out from a vertex, use up 2 edge
2. So the 1st & last vertices must have even degree.
3. But the 7 bridges graph has 4 vertizes of odd degree.

31.4 Euler's theorem

→ a trail that uses every edge of a graph exactly once = Euler trail

→ a graph with >2 odd degree vertizes has no Euler trail [must be exactly 2]

→ a graph with odd degree vertizes has no closed Euler trail.

Lecture 32: Trees

V1
⇒ consider as a graph

→ Tree must be a connected graph

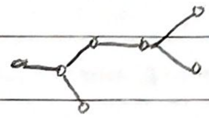
→ can get from any vertex to any other vertex

→ every edge is a bridge

→ right at limit of being disconnected

→ no subgraph, no cycle, no loop

eg:

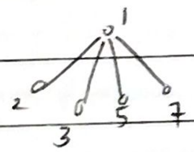


eg: which one are trees

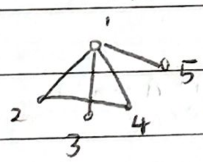
→ edge betw vertices m and n when m ≠ n,

either m either n or n divide m.

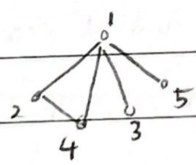
A. vertices 1, 2, 3, 5, 7



B. 1, 2, 3, 4, 5

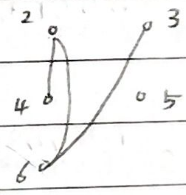


or



[cycle]

C. 2, 3, 4, 5, 6



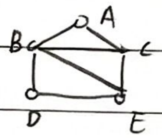
[disconnected graph]

→ BFS & DFS to construct a spanning tree of a connected graph.

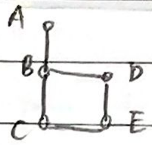
Date: 23/5/2022.

33.3 Breadth first algorithm " 1st in - 1st out " [go from 1st place I visited]

"queue"

eg: G = 

root vertex A.

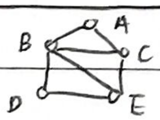
Q: 

Step	Q	T
1.	A	
2.	AB	
3.	ABC	
4.	BC	(A come out)
5.	BCD	
6.	BCDE	
7.	CDE	(B come out)
8.	DE	everything connected to C alrdy inside, C come out
9.	E	

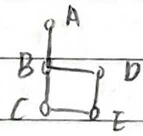
Q	T
A	
AB	
B	
BC	
BCD	
CD	
CDE	
DE	
E	

33.4 Depth first algorithm "stack" [go from last place I visited]

" first in - last out "

G = 

root vertex A.

Q: 

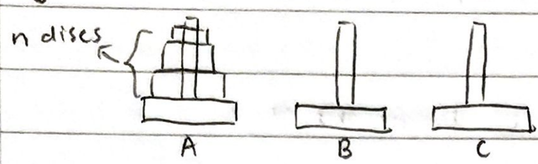
S	T
A	
AB	
ABC	
ABCE	
ABCED	
ABCE	
ABC	
AB	
A	

S	T
A	
AB	
ABC	
ABCE	
ABCED	
ABCE	
ABC	
AB	

Lecture 9: mathematical induction [Weak Induction]

→ to proof something, way of reasoning & thinking

eg:



$P(n)$: Any stack of n discs can be tangled, $n \in \mathbb{Z}$

① <u>Base step</u>: For a stack of 1 discs, it can be done

$\Rightarrow$ $P(1)$ is true

② <u>Inductive step</u>

n	1	2	3	4	5	....
$P(n)$	T	T	T	T	T	....

[expect to be true for all cases]

→ Assume that $P(k)$ is true [assuming that a stack of k discs can be moved]

→ We have to prove that $P(k+1)$ will be true also

$P(k) \rightarrow P(k+1)$

→ We have to prove that a stack of $(k+1)$ discs can also be moved.

Conclusion: [if you can do k, then you can do $(k+1)$ case]

since $P(1) \equiv T$ {Base - step}

$P(1) \rightarrow P(2)$ {Inductive step $k=1$}

$P(2) \rightarrow P(3)$ { .... $k=2$}

⋮

$\boxed{\therefore\ P(n) \equiv T \text{ for all } n \geq 1}$

Generally

Base Step = base on the ques

Inductive Step = (k) & (k + 1) cases

Conclusion = Since <base step>,

<IS>, ...

<conclude from ques> #

Weak Induction

Since $P(1)$ is true	Base step
$P(1) \rightarrow P(2)$	IS
$P(3) \rightarrow P(3)$	IS
$\vdots$	
$\therefore P(n) \equiv T$	all k

Strong Induction

Since $P(1) \wedge P(2)$ are true

$P(1) \wedge P(2) \rightarrow P(3)$ IS $k=2$

$P(1) \wedge P(2) \wedge P(3) \rightarrow P(4)$ IS $k=3$

$P(1) \wedge P(2) \wedge P(3) \wedge P(4) \rightarrow P(5)$ IS $k=4$

$\vdots$

$\therefore P(n)$ is true

(1) P(n) : Every a_n is even for $n \geq 0$,

whr $a_0 = 2$, $a_1 = 6$, $a_n = a_{n-1} + a_{n-2}$

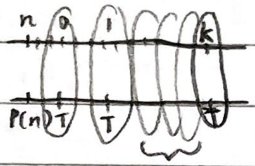
(2) Base step: Given $a_0 = 2$, & 2 is indeed even

$\Rightarrow P(0) \equiv T$

and $a_1 = 6$, and 6 is indead even

$\Rightarrow P(1) \equiv T$

(3) Inductive step:



Want to prove this

Assume $P(0) \land P(1) \land P(2) \ldots \land \ldots P(k) \rightarrow P(k+1)$

Assume that $P(n)$ is true for all $n \leq k$, $n \in \mathbb{Z}$, $k \in \mathbb{Z}$, $k > 1$

WHTP $P(k+1)$ is true

→ is $a_k, a_{k-1}, a_{k-2}, a_{k-3} \ldots a_n$ all are even #

→ WHTP a_{k+1} is an even #

Now $a_{k+1} = a_k + a_{k-1}$ from recursive relationship

$= 2M + 2N$ $\quad P(k) \land P(k-1)$ from assumption is true

$= 2[M+N]$ $\quad M, N \in \mathbb{Z}$

$a_{k+1} = 2W$ $\quad W = M+N \in \mathbb{Z}$

$\Rightarrow P(k+1) \equiv T$

(4) Conclusion: Since $P(0) \land P(1) \equiv T$ (BS)

$P(0) \land P(1) \rightarrow P(2)$ (IS, $k=1$)

$P(0) \land P(1) \land P(2) \rightarrow P(3)$ (IS, $k=2$)

$\vdots$

$\therefore P(n) \equiv T$ for all $n \geq 0$