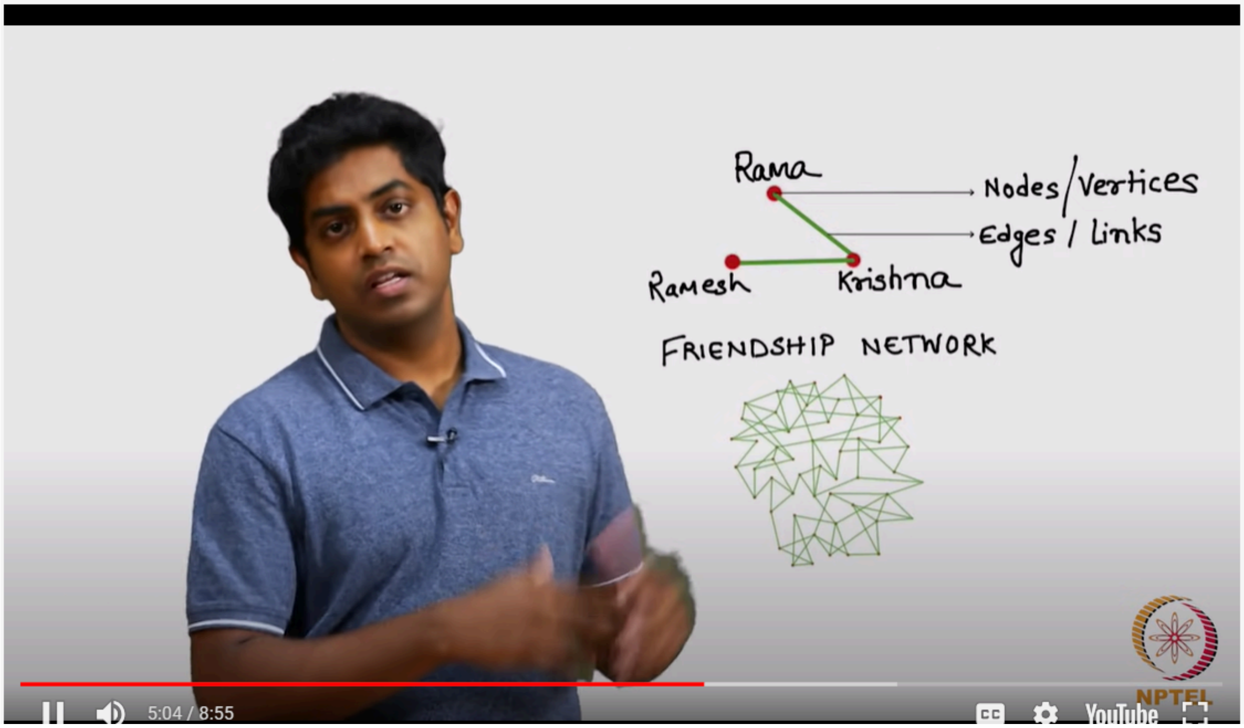
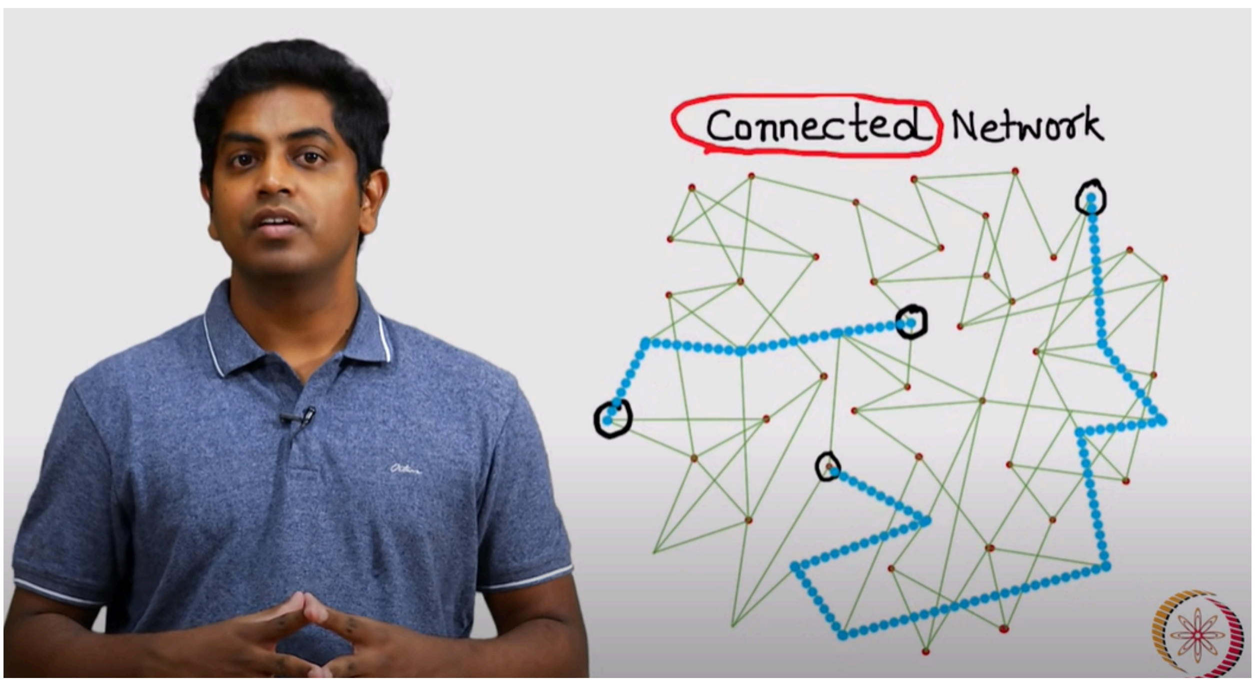
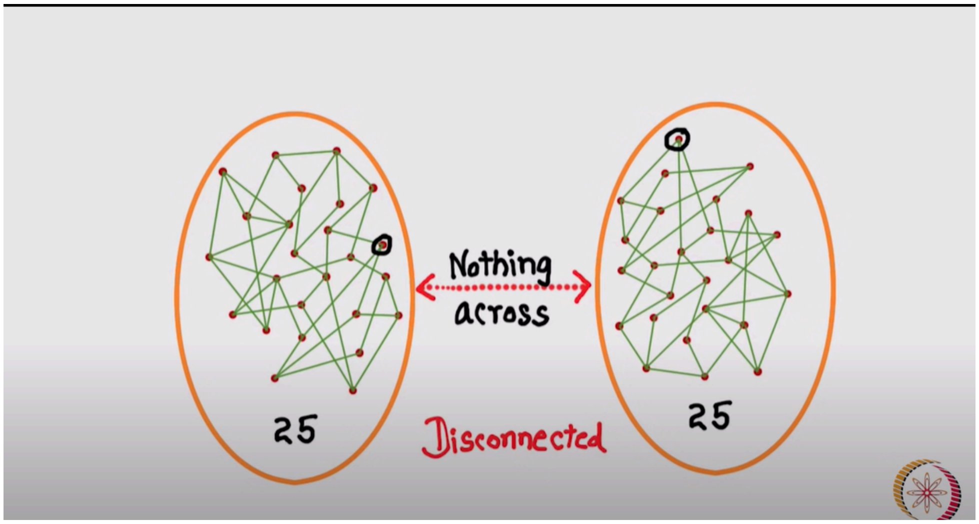
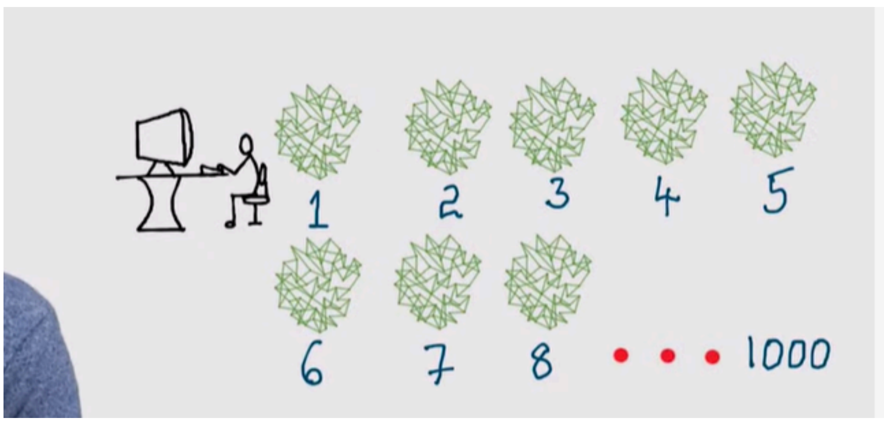
Lecture 01 - Introduction (9 min)

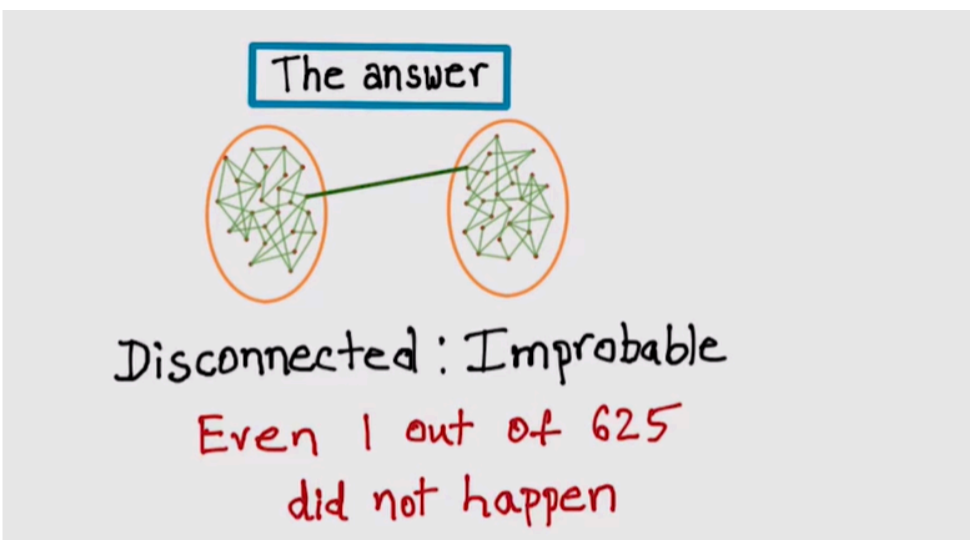






This did not happen in real. When u take 50persons and make 3frnds for each person. Everytime its a connected graph.





Lecture 03 - Introduction to Python-1 (21 min)

```
In [1]: a=2

In [2]: b=5

In [3]: c='sudarshan'

In [4]: a,b=b,a

In [5]: print a
5

In [6]: print b
2

In [7]: a,b=b,a

In [8]: print a
2
```

```
In [11]: for i in range(10):
    ...:     print i
    ...:
0
1
2
3
4
5
6
7
8
9
```

```
In [12]: k=10

In [13]: while (k<=20):
    ...:     print k
    ...:     k=k+1
    ...:
10
11
12
13
14
15
16
17
18
19
20

In [14]: print k
21

In [15]: if (k>20):
    ...:     print "The value of k is greater than 20"
    ...:
The value of k is greater than 20

In [16]: if (k>25):
    ...:     print "something something"
    ...: else:
    ...:     print "the
```

NPTEL

```
In [18]: import random

In [19]: random.randrange(1,10)
Out[19]: 4

In [20]: random.randrange(1,10)
Out[20]: 5
```

```
In [56]: random.randrange?
Signature: random.randrange(start, stop=None, step=1, _int=<type 'int'>, _maxwidth=9007199254740992L)
Docstring:
Choose a random item from range(start, stop[, step]).
```

```
In [62]: random.random()
Out[62]: 0.1938257517766626

In [63]: random.random()
Out[63]: 0.16229121181178352
```

no. Between 0 and 1

```
In [77]: random.randint?
Signature: random.randint(a, b)
Docstring:
Return random integer in range [a, b], including both end points.

File:      /usr/local/Cellar/python/2.7.13/Frameworks/Python.framework/Versions/2.7/lib/python2.7/random.py
Type:      instancemethod

In [78]: random.randint(1,5)
Out[78]: 2
```

LISTS

```
In [2]: L=[]

In [3]: L.append(2)

In [4]: print L
[2]

In [5]: L.append(10)

In [6]: print L
[2, 10]

In [7]: L.append(100)

In [8]: print L
[2, 10, 100]
```

```
In [14]: L.sort()

In [15]: print L
[2, 7, 10, 14, 77, 100]

In [16]: L.reverse()

In [17]: print L
[100, 77, 14, 10, 7, 2]

In [18]: L.
          L.append   L.index    L.remove
          L.count    L.insert   L.reverse
          L.extend   L.pop      L.sort
```

Press L.Tab to see all options

```
In [21]: import random

In [22]: for i in range(100):
    ...:     L.append(random.random())
```

```
In [27]: for i in range(40):
    ...:     L.append(random.randint(1,365))
    ...:

In [28]: print L
[210, 250, 94, 201, 187, 259, 158, 323, 236, 185, 100, 75, 139, 46, 198,
192, 145, 357, 256, 39, 241, 222, 22, 209, 101, 173, 353, 316, 339, 10, 2
5, 359, 89, 348, 175, 70, 111, 230, 189, 267]
```

```
In [43]: Birthday Paradox
```

If you survey a random group of just 23 people there is actually about a 50–50 chance that two of them will have the same birthday - during lecture he said 60+ people

FUNCTIONS

```
In [109]: def sumup():
     ...:     a=random.random()
     ...:     b=random.random()
     ...:     return a+b
     ...:

In [110]: sumup()
Out[110]: 1.1412489051246113
```

```
In [118]: !vi sudarshan.py
```

OPENING ANOTHER OUTSIDE EDITOR

```
import random
def sum3rand():
    a=random.random()
    b=random.random()
    c=random.random()
    return a+b+c
```

.py file

```
In [120]: !cat sudarshan.py
import random
def sum3rand():
    a=random.random()
    b=random.random()
    c=random.random()
    return a+b+c
```

CAT is used for seeing the contents of file

```
In [121]: import sudarshan

In [122]: sudarshan.sum3rand()
Out[122]: 2.472613360995443
```

invoking the file in which function is written

Update the file:

```
import random
def sum3rand():
    a=random.random()
    b=random.random()
    c=random.random()
    return a+b+c

def sumkrand(k):
    ans=0
    for i in range(k):
        ans=ans+random.random()
    return ans
```

Reload it

```
In [141]: reload(sudarshan)
Out[141]: <module 'sudarshan' from 'sudarshan.py'>

In [142]: sudarshan.sumkrand(10)
Out[142]: 4.575307276426816

In [143]: sudarshan.sumkrand(10)
Out[143]: 5.2256366698481544

In [144]: sudarshan.sumkrand(100)
```

Update the file to contain a helper line in ''' '''

```python
def sumkrand(k):
    '''This function takes k as input and adds random numbers
    between 0 and 1, k times and then outputs the answer'''
    ans=0
    for i in range(k):
        ans=ans+random.random()
    return ans
```

```
In [162]: reload(sudarshan)
Out[162]: <module 'sudarshan' from 'sudarshan.py'>

In [163]: sudarshan.sumkrand?
Signature: sudarshan.sumkrand(k)
Docstring:
This function takes k as input and adds random numbers
between 0 and 1, k times and then outputs the answer
File:      ~/nptel/sudarshan.py
Type:      function
```

DICTIONARY

- Stored in hash tables - faster than lists
- Elements in dict are not accessed with index, they are accessed by keys - d[0] ->error
- Order of elements is not imp

```
In [1]: d = {'annie':25, 'yoyo':35, 'sid':65}

In [2]: d
Out[2]: {'annie': 25, 'sid': 65, 'yoyo': 35}
```

```
KeyError                                  Traceback (most recent call last)
<ipython-input-3-17371c6688f6> in <module>()
----> 1 d[0]

KeyError: 0

In [4]: d['annie']
Out[4]: 25

In [5]: d[0]
---------------------------------------------------------------------------
```

Updating value and adding element -

```
In [6]: d
Out[6]: {'annie': 25, 'sid': 65, 'yoyo': 35}

In [7]: d['sid'] = 75

In [8]: d
Out[8]: {'annie': 25, 'sid': 75, 'yoyo': 35}

In [9]: d['john'] = 15

In [10]: d
Out[10]: {'annie': 25, 'john': 15, 'sid': 75, 'yoyo': 35}
```

Deleting an entry -

```
In [11]: del d['john']

In [12]: d
Out[12]: {'annie': 25, 'sid': 75, 'yoyo': 35}
```

```
In [13]: d.clear()
```

- remove all elements from dictionary but d exists

```
In [13]: del d
```

- delete the d from the memory

```
In [14]: d.has_key('annie')
Out[14]: True

In [15]: d.keys()
Out[15]: ['yoyo', 'annie', 'sid']

In [16]: d.values()
Out[16]: [35, 25, 75]
```

d.items() - give list of tuples having pair of key , value

```
In [17]: d.items()
Out[17]: [('yoyo', 35), ('annie', 25), ('sid', 75)]
```

```
In [18]: for i in d.items():
   ....:     print i
   ....:
('yoyo', 35)
('annie', 25)
('sid', 75)
```

Accessing by index in d.items() list -

```
In [19]: for i in d.items():
    print i[0], 'has age: ', i[1]
   ....:
yoyo has age:  35
annie has age:  25
sid has age:  75
```

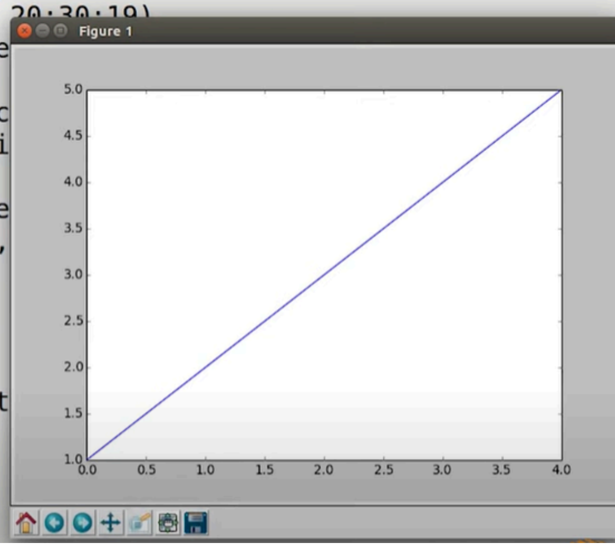
```
In [22]: d1 = {x:x**2 for x in range(11)}

In [23]: d1
Out[23]: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81,
10: 100}
```

```
In [24]: d1 = {x:x**2 for x in range(11) if x%2 == 0}

In [25]: d1
Out[25]: {0: 0, 2: 4, 4: 16, 6: 36, 8: 64, 10: 100}
```

PLOTTING

```
In [1]: import matplotlib.pyplot as plt
```

```
Python 2.7.6 (default, Oct 26 2016,
Type "copyright", "credits" or "lice

IPython 1.2.1 -- An enhanced Interac
?         -> Introduction and overvi
%quickref -> Quick reference.
help      -> Python's own help syste
object?   -> Details about 'object',

In [1]: import matplotlib.pyplot as

In [2]: plt.plot([1,2,3,4,5])
Out[2]: [<matplotlib.lines.Line2D at

In [3]: plt.show()
```



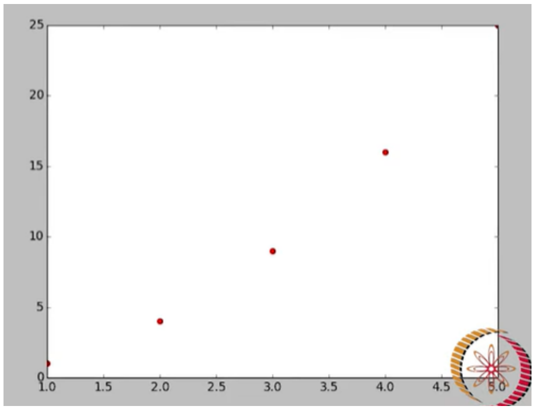
List items 1,2,3,4,5 are taken as Y values by default and x values are not given so matplotlib takes on its own

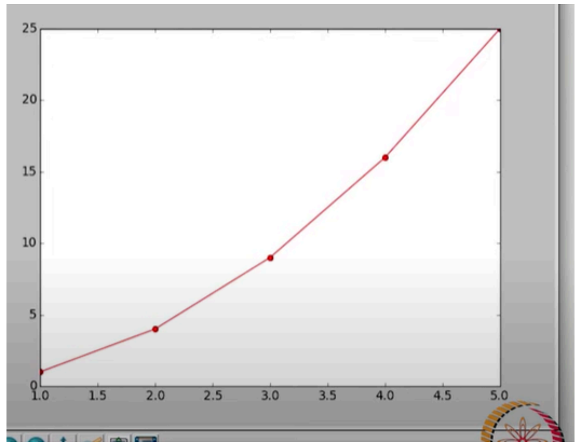
Passing both x and y values- 1st list - Xaxis, 2nd list - Yaxis

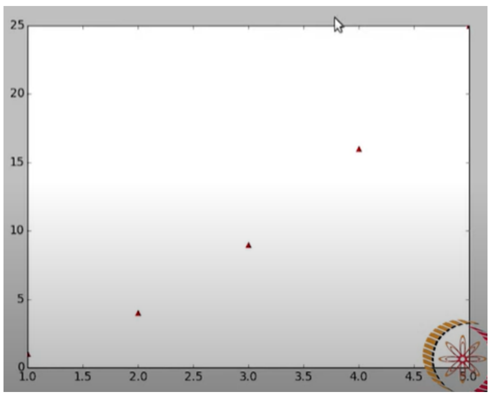
```
In [4]: plt.plot([1,2,3,4,5],[1,4,9,16,25])
Out[4]: [<matplotlib.lines.Line2D at 0x7f5f468cec50>]
```

```
In [6]: plt.plot([1,2,3,4,5],[1,4,9,16,25],'ro')
Out[6]: [<matplotlib.lines.Line2D at 0x7f5f4618ac50>]

In [7]: plt.show()

In [8]: plt.plot([1,2,3,4,5],[1,4,9,16,25],'ro-')
Out[8]: [<matplotlib.lines.Line2D at 0x7f5f45fd6f50>]
```

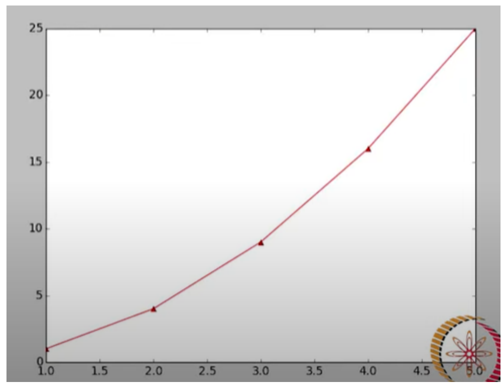
ro -> gives dots only, ro- -> gives lines and dots both (in red)


25
20
15
10
5
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

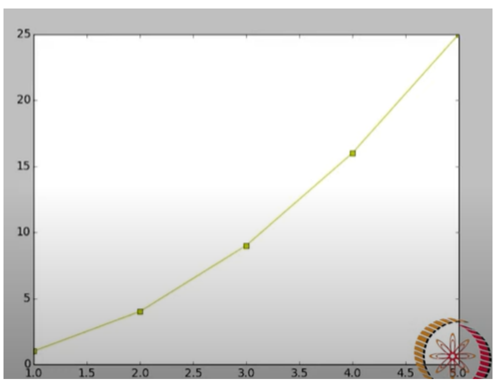

25
20
15
10
5
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


25
20
15
10
5
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

r^

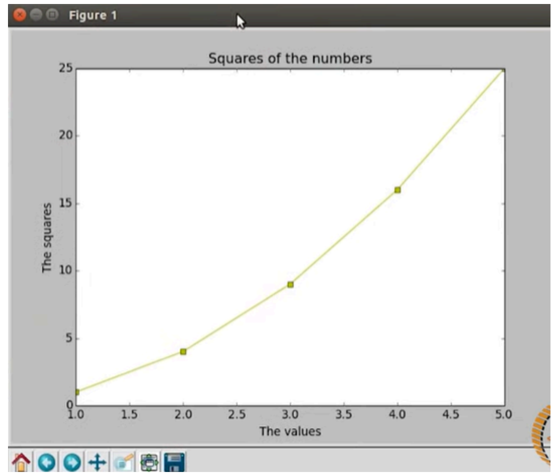

25
20
15
10
5
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

r^-


25
20
15
10
5
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

’ys-’

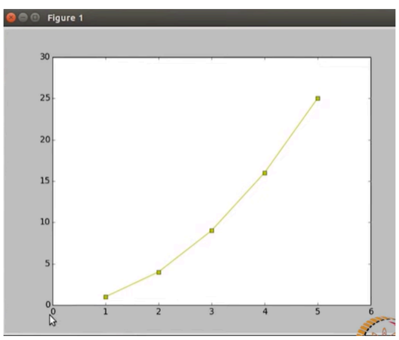
```
In [16]: plt.title('Squares of the numbers')
Out[16]: <matplotlib.text.Text at 0x7f5f45d07210>

In [17]: plt.xlabel('The values')
Out[17]: <matplotlib.text.Text at 0x7f5f45e2fbd0>

In [18]: plt.ylabel('The squares')
Out[18]: <matplotlib.text.Text at 0x7f5f45e5fad0>

In [19]: plt.plot([1,2,3,4,5],[1,4,9,16,25],'ys-')
Out[19]: [<matplotlib.lines.Line2D at 0x7f5f45d66390>]

In [20]: plt.show
```



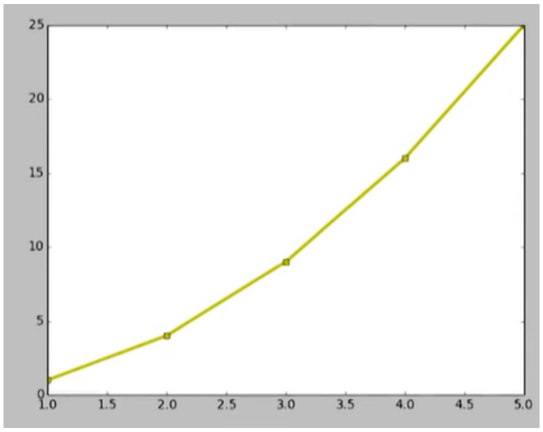

```
In [21]: plt.xlabel('The values', fontsize = 18, color = 'red')
Out[21]: <matplotlib.text.Text at 0x7f5f45d68350>
```


```
In [24]: plt.axis([0,6,0,30])
Out[24]: [0, 6, 0, 30]
```

1st value - start of Xaxis , 2nd - end of X , 3rd - start of Y , 4th =end of Y

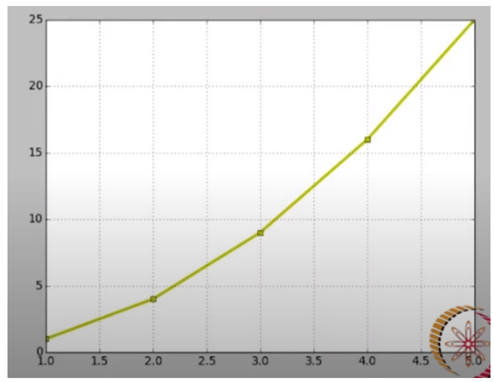



```
In [27]: plt.plot([1,2,3,4,5],[1,4,9,16,25],'ys-', linewidth = 3.0)
Out[27]: [<matplotlib.lines.Line2D at 0x7f5f45ad5910>]
```



Set grid = true to see grid in plot which is false by default

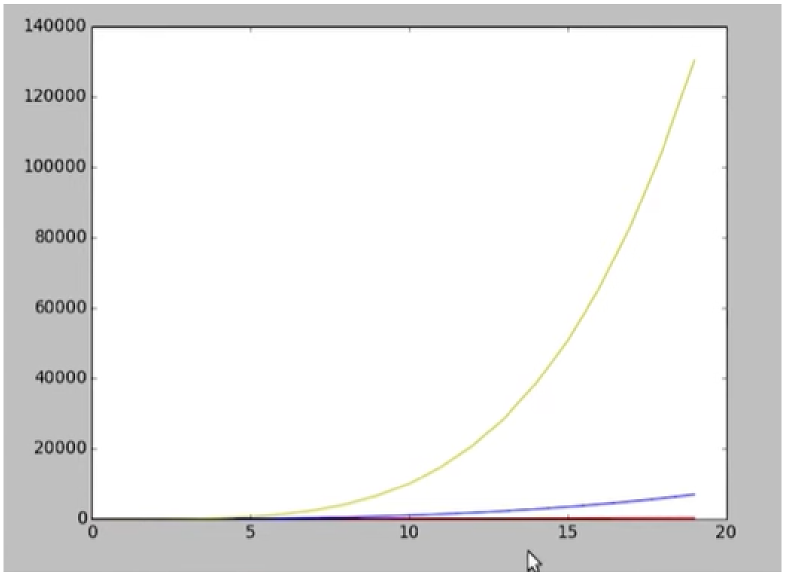
```
[n [29]: plt.grid(True)

[n [30]: plt.plot([1,2,3,4,5],[1,4,9,1(
)ut[30]: [<matplotlib.lines.Line2D at (

[n [31]: plt.show()
```




```
In [32]: x = [i for i in range(20)]

In [33]: plt.plot(x, x**2, 'r-', x, x**3, 'b-', x, x**4, 'y-')
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-33-724b24d2d8a2> in <module>()
----> 1 plt.plot(x, x**2, 'r-', x, x**3, 'b-', x, x**4, 'y-')

TypeError: unsupported operand type(s) for ** or pow(): 'list' and 'int'
```

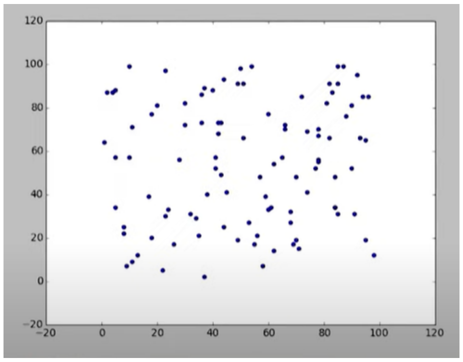
Error occurred bcz exp on list is not supported
So we use a **numpy array** to allow this.\

```
In [34]: import numpy

In [35]: x = numpy.array(x)

In [36]: plt.plot(x, x**2, 'r-', x, x**3, 'b-', x, x**4, 'y-')
Out[36]:
[<matplotlib.lines.Line2D at 0x7f5f45937090>,
 <matplotlib.lines.Line2D at 0x7f5f45937310>,
 <matplotlib.lines.Line2D at 0x7f5f459379d0>]
```



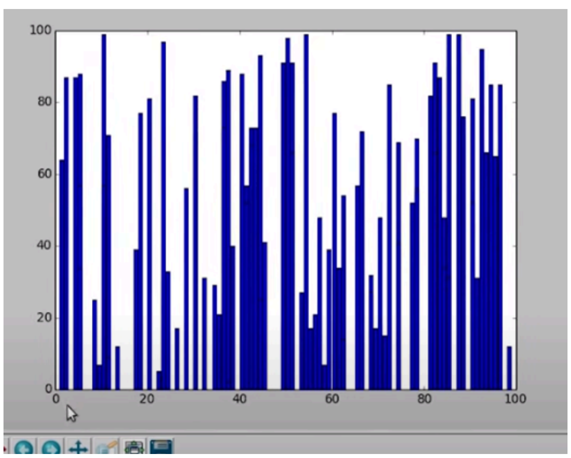
SCATTER PLOT

```
In [38]: import random

In [39]: x = [random.randint(1,100) for i in range(100)]

In [40]: y = [random.randint(1,100) for i in range(100)]

In [41]: plt.scatter(x,y)
Out[41]: <matplotlib.collections.PathCollection at 0x7f5f4584fc50>

In [42]: plt.show()
```



BAR CHART

```
5]: plt.bar(x,y)
6]: <Container object of

7]: plt.show()
```



Lecture 05 - Introduction to Networkx-1

```
In [13]: import networkx

In [14]: G=networkx.Graph()

In [15]: G.add_node(1)

In [16]: G.add_node(2)

In [17]: G.add_node(3)

In [18]: G.add_node(4)

In [19]: G.add_node(5)

In [20]: G.nodes()
Out[20]: [1, 2, 3, 4, 5]

In [21]: G.add_node(6)

In [22]: G.nodes()
Out[22]: [1, 2, 3, 4, 5, 6]
```

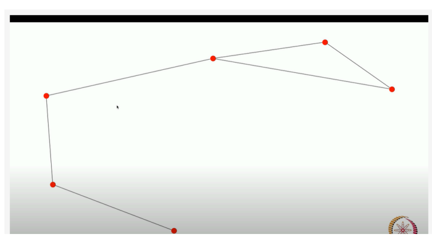
```
In [23]: G.add_edge(1,2)

In [24]: G.add_edge(1,3)

In [25]: G.add_edge(4,6)

In [26]: G.add_edge(5,4)
```

```
In [29]: G.edges()
Out[29]: [(1, 2), (1, 3), (2, 3), (4, 5), (4, 6)]
```

```
In [32]: import networkx as nx

In [33]: H=nx.Graph()
```

For visualising the graph - use matplotlib.pyplot

```
In [34]: G.nodes()
Out[34]: [1, 2, 3, 4, 5, 6]

In [35]: G.edges()
Out[35]: [(1, 2), (1, 3), (2, 3), (2, 6), (4, 5), (4, 6)]

In [36]: import matplotlib.pyplot as plt

In [37]: nx.draw(G)

In [38]: plt.show()
```



Here we have to display the labels in the graph

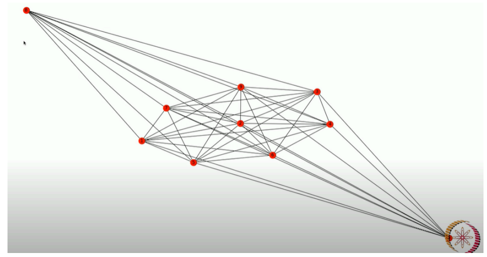
```
In [25]: nx.draw(G,with_labels=1)

In [26]: plt.show()
```

Built-in funcns

```
In [8]: Z=nx.complete_graph(10)
```

- generates a complete graph with 10 nodes

```
In [15]: nx.draw(Z,with_labels=1)

In [16]: plt.show()
```

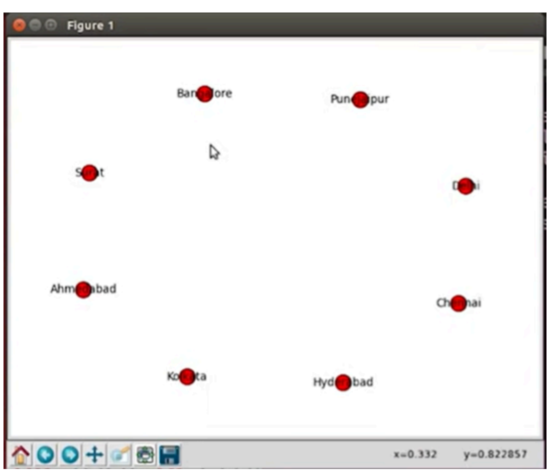


```
In [4]: G=nx.gnp_random_graph(20,0.5)

In [5]: nx.draw
```

- prob of edge between 2 vertices is 0.5

MODELING ROAD NETWORK FOR INDIA'S CITIES

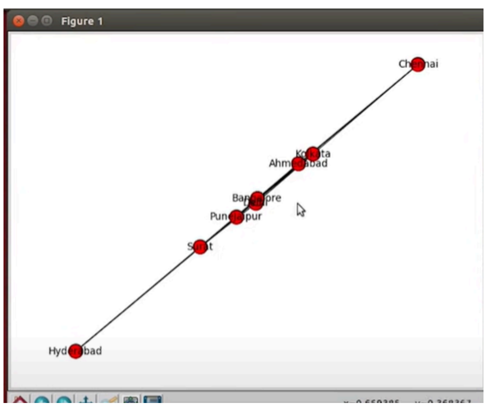
```
import networkx as nx
import matplotlib.pyplot as plt

G=nx.Graph()    #undirected graph

#G=nx.DiGraph() #Directed graph

city_set=
['Delhi','Bangalore','Hyderabad','Ahmedabad','Chennai','Kolkata',
'Surat','Pune''Jaipur']

for each in city_set:
        G.add_node(each)

nx.draw(G)
plt.show()
```



Making a cost array for the edges

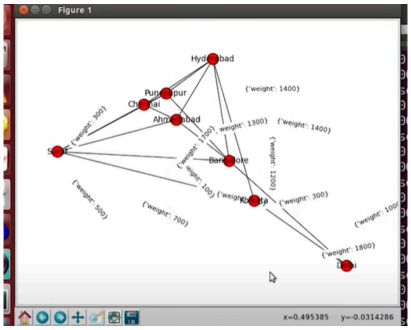
```
costs=[]
value=100
while (value<=2000):
        costs.append(value)
        value=value+100
print costs
```

```
#We are going to add 16 edges to this network
while(G.number_of_edges()<16):
        c1=random.choice(G.nodes())
        c2=random.choice(G.nodes())
        if c1!=c2 and G.has_edge(c1,c2)==0:
                w=random.choice(costs)
                G.add_edge(c1,c2,weight=w)

nx.draw(G)
plt.show()
```



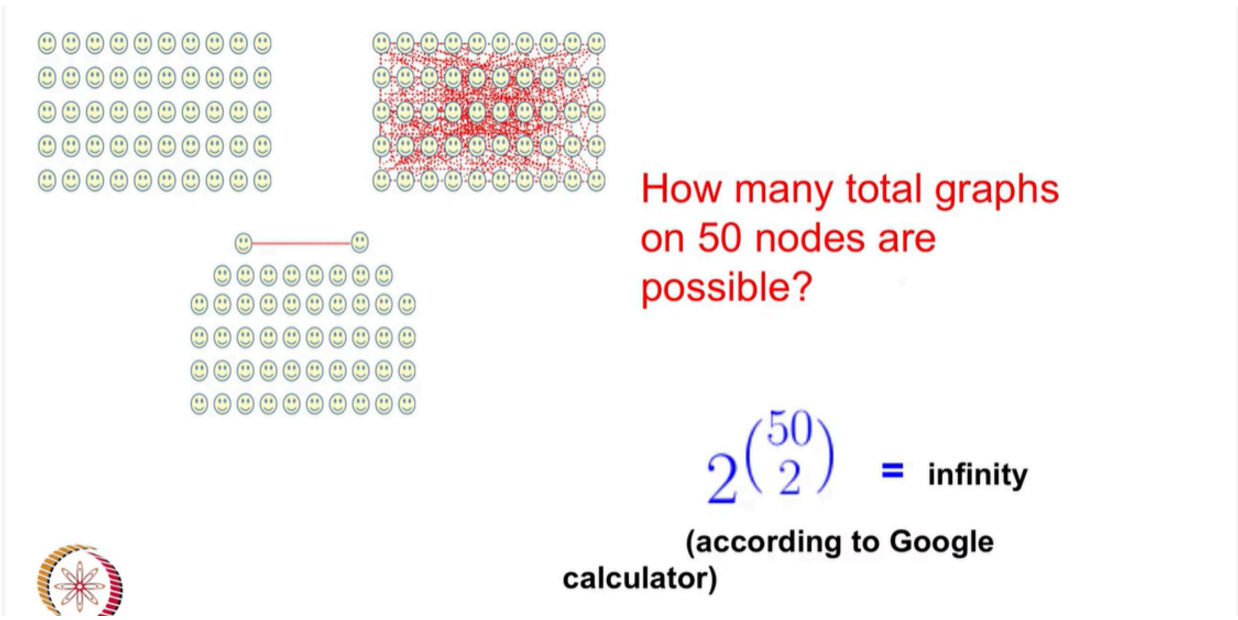
For better layout, change the layout and write edge weights -

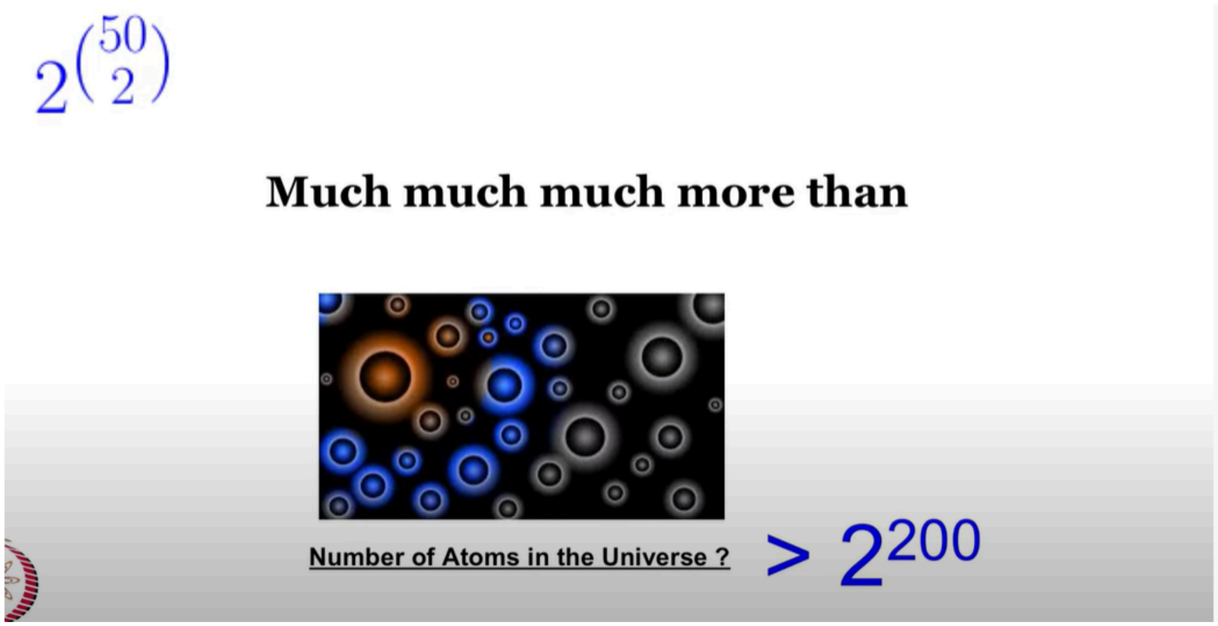
```
                G.add_edge(c1,c2,weight=w)
pos=nx.circular_layout(G)
nx.draw(G)
nx.draw_networkx_edge_labels(G,pos)
plt.show()
```



—-------------------COMPLETE THIS LECTURE —--

Lecture 07 - Social Networks: The Challenge (4 min)

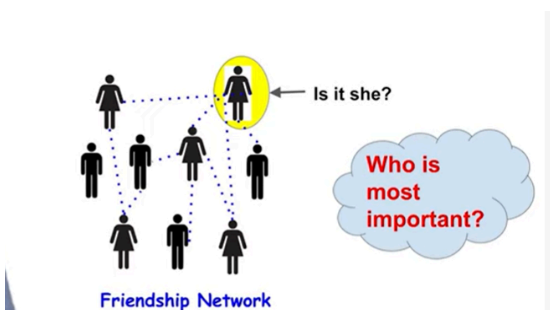


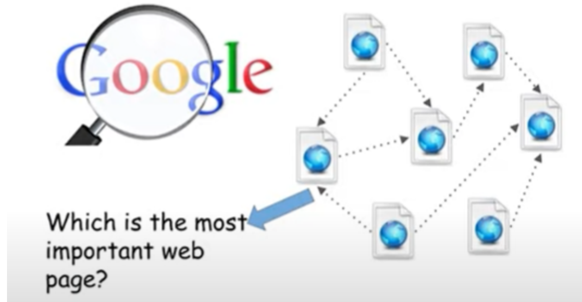


STILL WE SEE

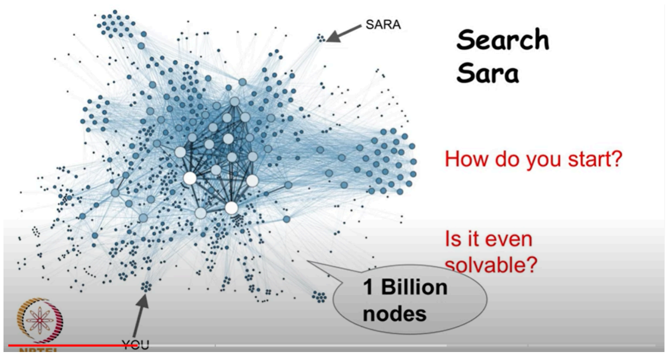
PROPERTIES -

1. Lecture 08 - Google Page Rank (2 min)

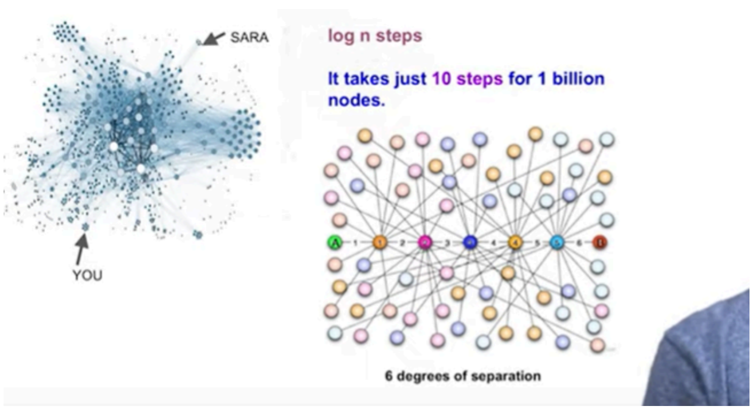




2. Lecture 09 - Searching in a Network (2 min)



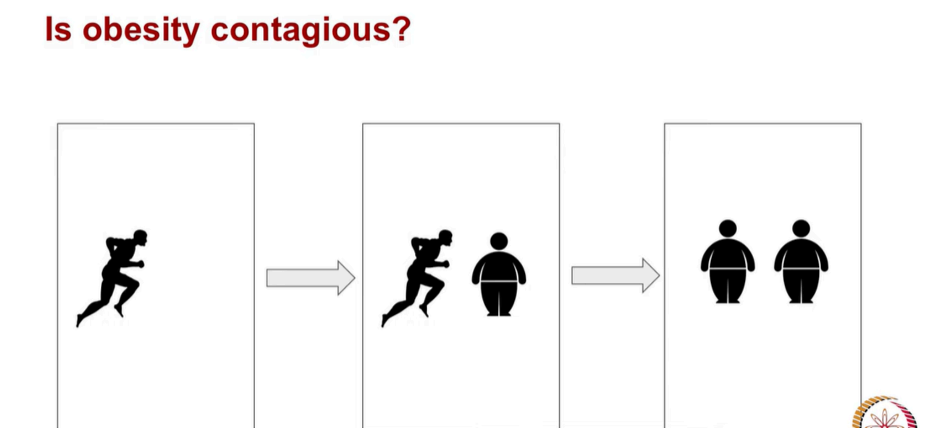
SMALL WORLD PHENOMENA

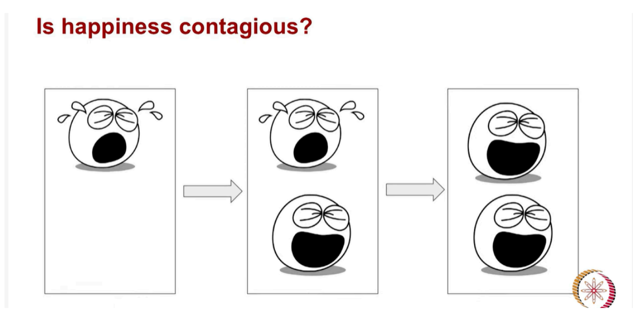


3. Lecture 10 - Link Prediction (2 min)

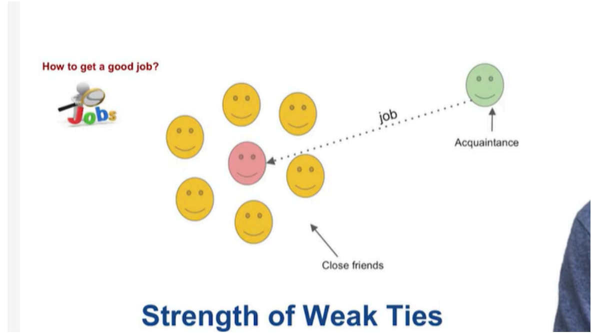
4. Lecture 11 - The Contagions (2 min)

IF YOUR FRND'S FRND IS HAPPY => YOU WILL BE HAPPY

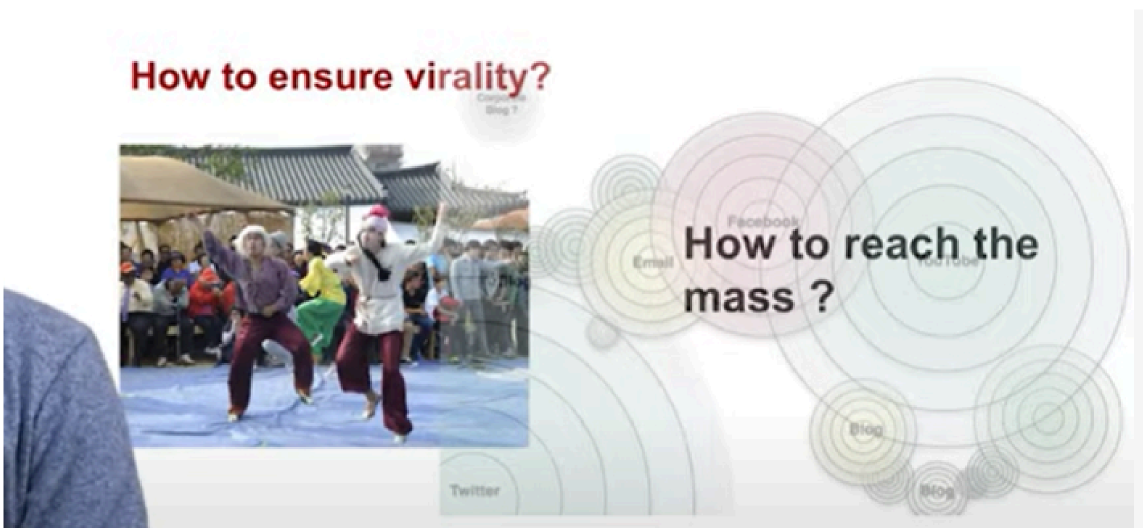




5. Lecture 12 - Importance of Acquaintances (1 min)

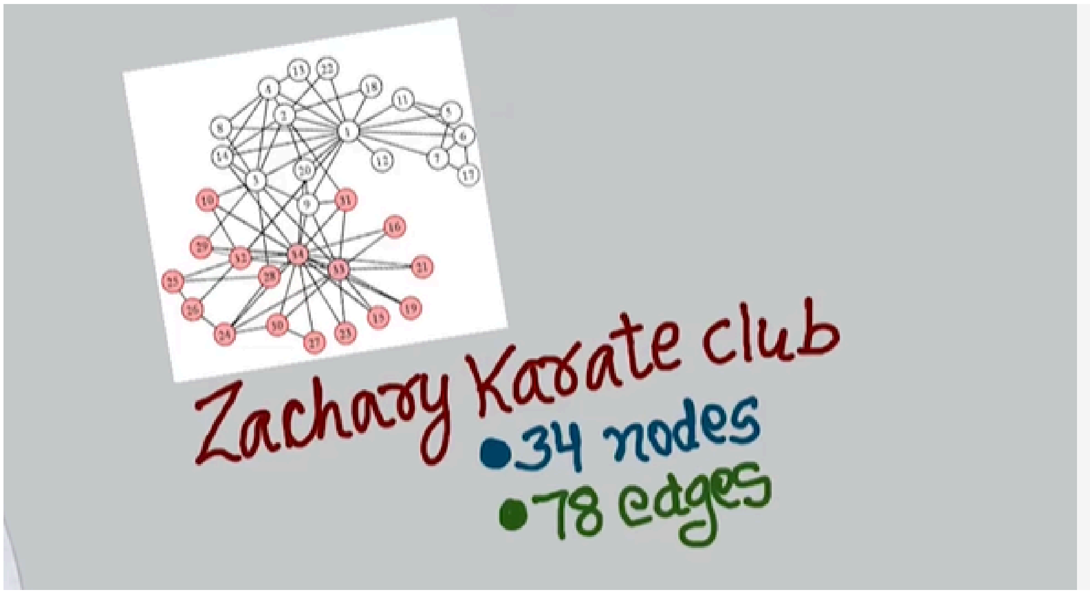


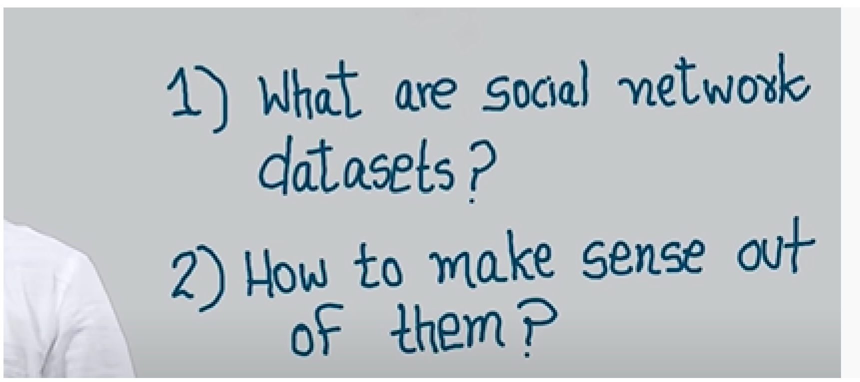
6. Lecture 13 - Marketing on Social Networks (2 min)



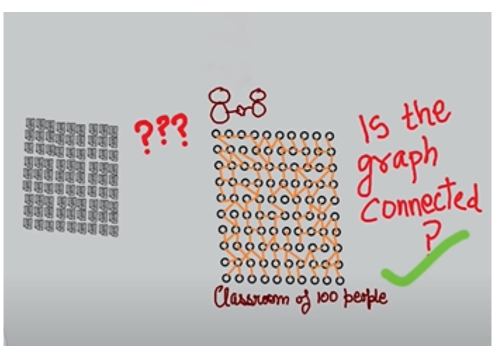
WEEK 2

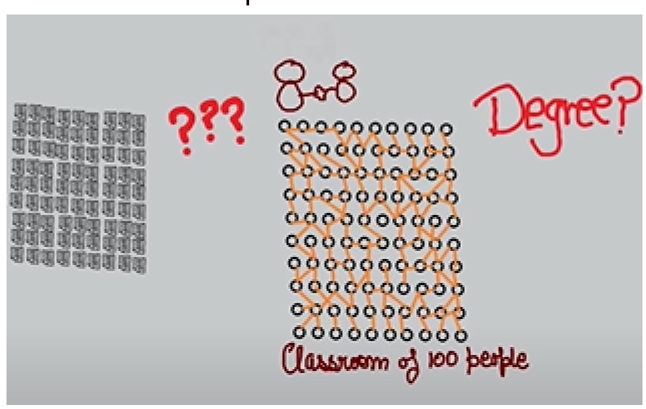
Lecture 14 - Introduction to Datasets

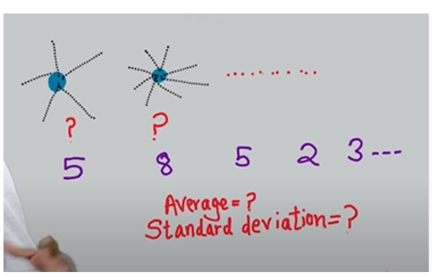




When u have data about connections u can ask various questions -



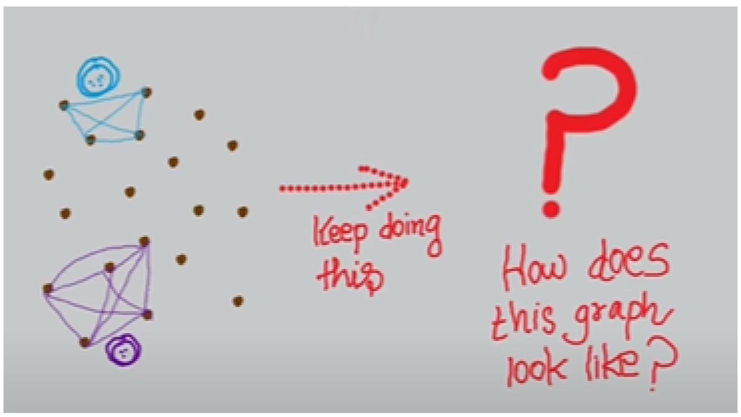




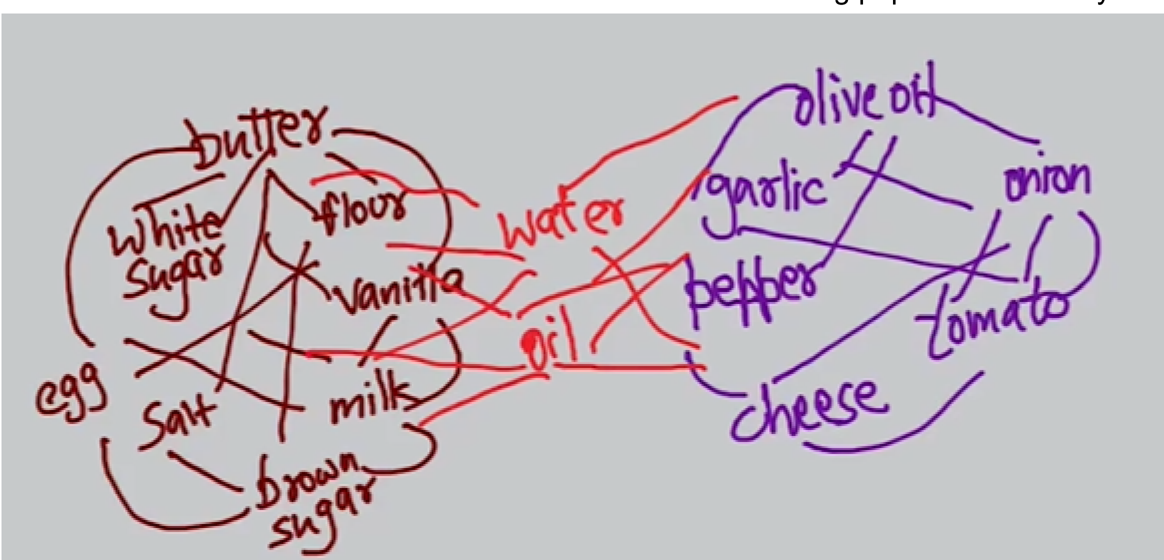
for degree of all the nodes

see random datasets and do this

Lecture 15 - Ingredients Network
Take all the ingredients in the world, put an edge between 2 if they are together part of a dish

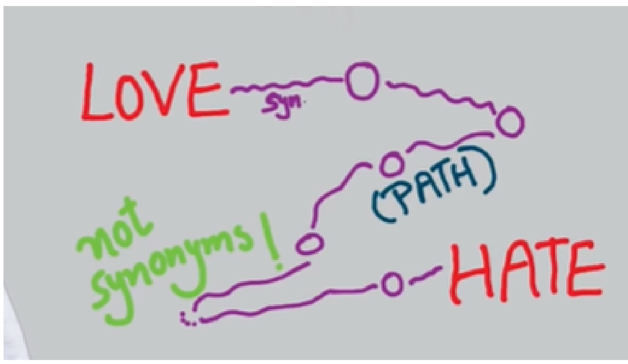


Looking popular dishes only -

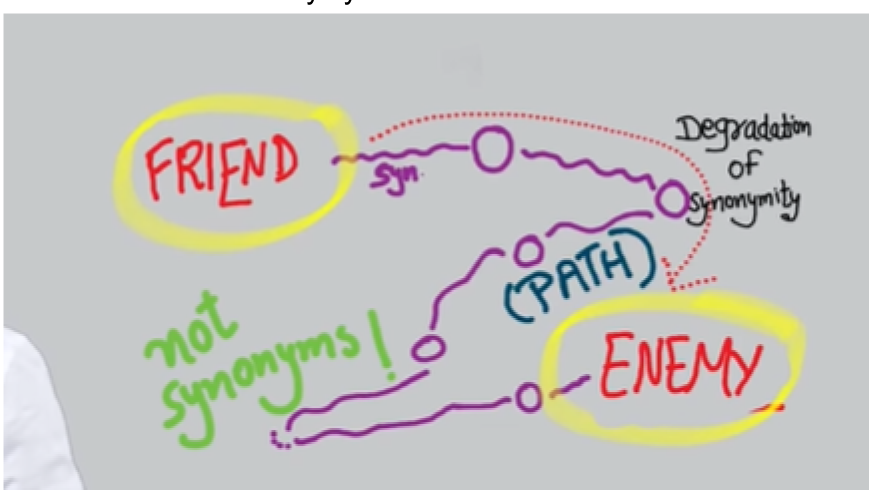


We observe strong communities in the ntwrk
Community structure - can give an overall hint about the ntwrk

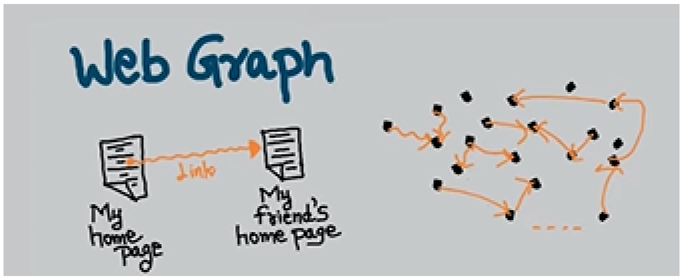
Lecture 16 - Synonymy Network
All words in english, put edge if they are synonyms

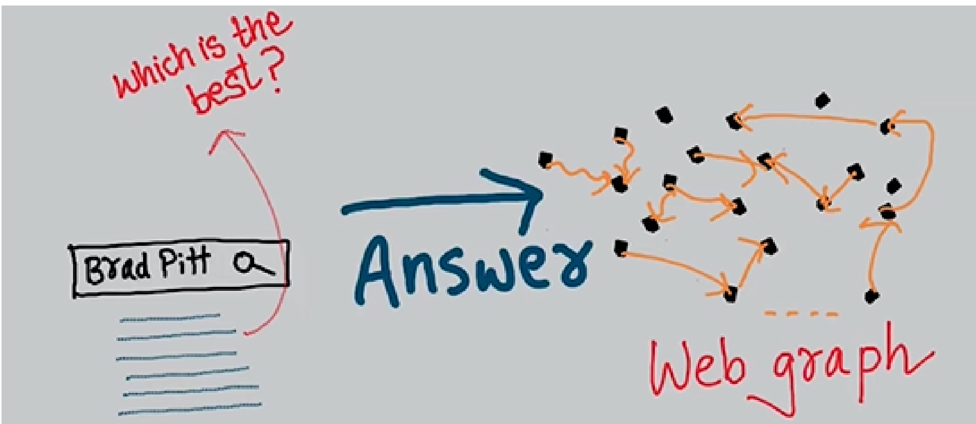


there exists a path from LOVE to HATE
Denotes - its not actually symmetric

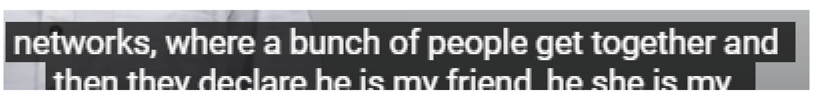


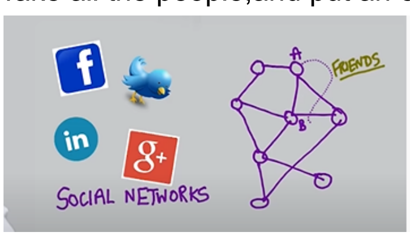
There is a degradation of synonymity

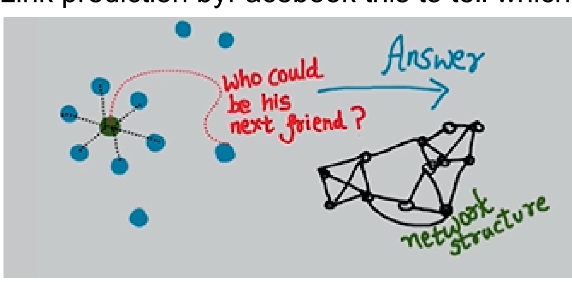
Lecture 17 - Web Graph
Pages connected with hyperlinks if there is a link from one page to another


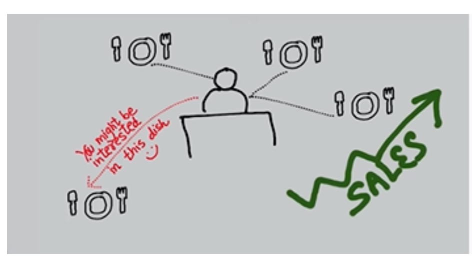
Useful in google to tell which is the best page for a particular keyword


Lecture 18 - Social Network Datasets

Take all the people,and put an edge if they are friendlists with each other


Link prediction byFacebook this to tell which should be the next frnd - most probable next frnd




Lecture 19 - Datasets: Different Formats

Real-World Network Data Sets

- Friendship Network
- Road Network
- Email Network
- Citation Network
- Collaboration / Co-authorship Network

Examples of Networks

N= people, edge = if frndship (undirected
N = Cities , E =if road exists b/w(undirected

- Directed. N= people , E = if email sent
- directed- if one ppr uses citation from other
- scientist collaborate with other scientist for particular work

Network Data Sets' Formats

- CSV
- GML
- Pajek Net
- GraphML
- GEXF

CSV FORMAT

> CSV: Comma Separated Values
> Extension: *.txt or *.csv
> - Edgelist
> - Adjacency List (Adjlist)

CSV FORMAT - Edgelist

```
0 344
0 345
0 346
0 347
1 48
1 53
1 54
1 73
1 88
1 92
```

edge from 0 - 344 , source and dest
You can also add weights for an edge

```
1 2 0.3
3 4 0.1
5 7 0.2
4 7 0.8
```

Limitation of edgelist format -

the edge list format carries and that you cannot add non numeric weights to the to the edges. You

cannot change the labels. You cannot change any , you cannot add an attribute to the nodes and

edges so that is a limitation that this format carries. So there is a trade off. So you can see

CSV FORMAT - AdjList
1st value = source node
Subsequent value tells the edges / adjacent vertices

```
1 2 5 7
2 4 6
3 1 4 7
4 6 2 3
5 7 2
6 1
7 2 8 4
8 9
9 1 7 4 8
```

as I already told youyou this format does not give you much flexibility, for example, you just cannot

add any sort ofum attribute or label to the nodes or edges. If we want to add some color attribute

to the nodes or edges you cannot do that. So it all depends on your requirement. Lets go to the

GML Format : Graph Modeling Language

```
graph
[
  node
  [
   id A
  ]
  node
  [
   id B
  ]
  node
  [
   id C
  ]
   edge
  [
   source B
   target A
  ]
  edge
  [
   source C
   target A
  ]
]
```

Write all the nodes and edges like above
Provides flexibility in providing labels, is not complex.

GML Format with Labels

```
graph
[
  node
  [
   id A
   label "Node A"
  ]
  node
  [
   id B
   label "Node B"
  ]
  node
  [
   id C
   label "Node C"
  ]
   edge
  [
   source B
   target A
   label "Edge B to A"
  ]
  edge
  [
   source C
   target A
   label "Edge C to A"
  ]
]
```

GML Format with Attributes

```
graph
[ hierarchic  1
  directed  1
  node
  [ id  0
    graphics
    [ x 200.0
      y 0.0
    ]
  ]
  node
  [ id  1
    graphics
    [ x 425.0
      y 75.0
    ]
  ]
  edge
  [ source  1
    target  0
    LabelGraphics
    [
        text  "Happy New Year!"
    ]
  ]
]
```

Pajek Net Format : Uses .NET extension

```
*Vertices 82670
1 "entity"
2 "thing"
3 "anything"
4 "something"
5 "nothing"
6 "whole"
```

```
*arcs
4244 107
4244 238
4244 4292
4247 107
4248 1
4248 54
```

.PAJ or .NET Extensions
82670 vertices
Entity is label for node 1
Then comes edges => Arcs = source : target

Pajek Net Format with attributes

```
*arcs
4244 107 5
```

5 = weight assigned

Pajek Net Format

```
*Vertices 9
*Edges
1 2
1 9
2 9
2 3
2 8
3 8
3 4
4 5
4 7
5 7
5 6
6 4
```

GraphML Format: Uses .graphml extension

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
<graph id="G" edgedefault="undirected">
<node id="n0"/>
<node id="n1"/>
<edge id="e1" source="n0" target="n1"/>
</graph>
</graphml>
```

GraphML Format with attributes

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/20
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.
<key id="d0" for="node" attr.name="color" attr.type="string">
<default>yellow</default>
</key>
<key id="d1" for="edge" attr.name="weight" attr.type="double"/>
<graph id="G" edgedefault="undirected">
<node id="n0">
<data key="d0">green</data>
</node>
<node id="n1"/>
<node id="n2">
<data key="d0">blue</data>
</node>
<node id="n3">
<data key="d0">red</data>
</node>
<edge id="e0" source="n0" target="n2">
<data key="d1">1.0</data>
</edge>
<edge id="e1" source="n0" target="n1">
<data key="d1">1.0</data>
</edge>
<edge id="e2" source="n1" target="n3">
<data key="d1">2.0</data>
```

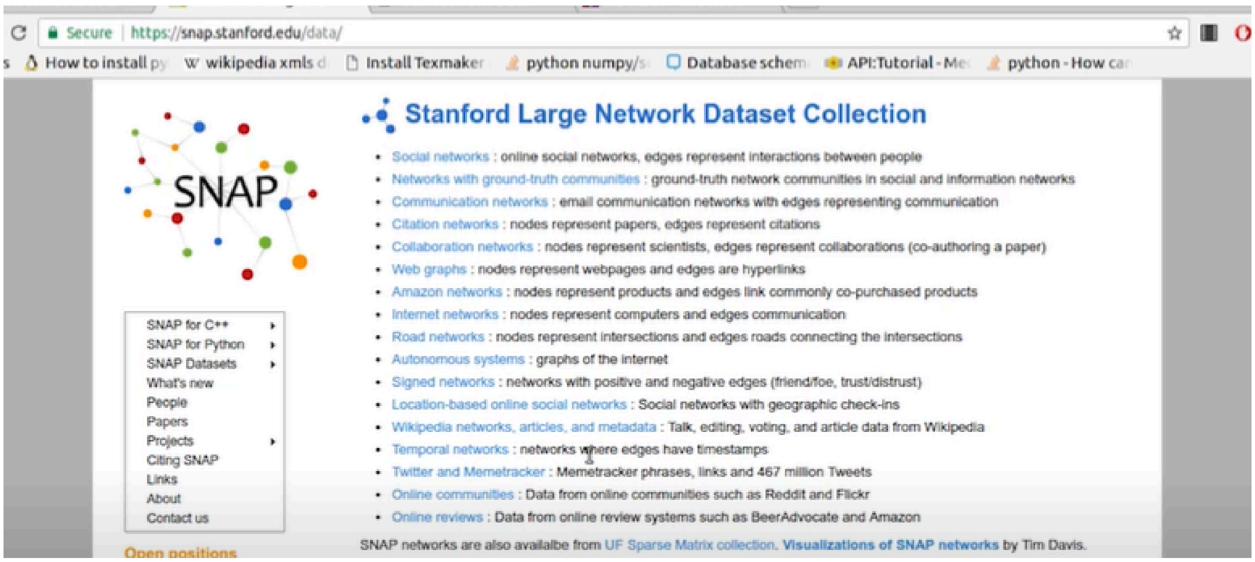

Key tag is used to add attributes like color , weight

GEXF Format : Graph Exchange XML Format

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">
    <meta lastmodifieddate="2009-03-20">
        <creator>Gexf.net</creator>
        <description>A hello world! file</description>
    </meta>
    <graph mode="static" defaultedgetype="directed">
        <nodes>
            <node id="0" label="Hello" />
            <node id="1" label="Word" />
        </nodes>
        <edges>
            <edge id="0" source="0" target="1" />
        </edges>
    </graph>
</gexf>
```

xml format this format was basically created
bygephi people gephi is an open source software

Lecture 20 - Datasets: How to Download?



Lecture 21 - Datasets: Analysing Using Networkx

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.read_edgelist('Data_sets/facebook_combined.txt')

print nx.info(G)

print nx.number_of_nodes(G)
print nx.number_of_edges(G)

print nx.is_directed(G)
```

```
anamika@anamika-Inspiron-5423:~/Desktop$ python data_sets.py
Name:
Type: Graph
Number of nodes: 4039
Number of edges: 88234
Average degree:  43.6910
4039
88234
False
anamika@anamika-Inspiron-5423:~/Desktop$
```

- For reading a pajek file in the graph

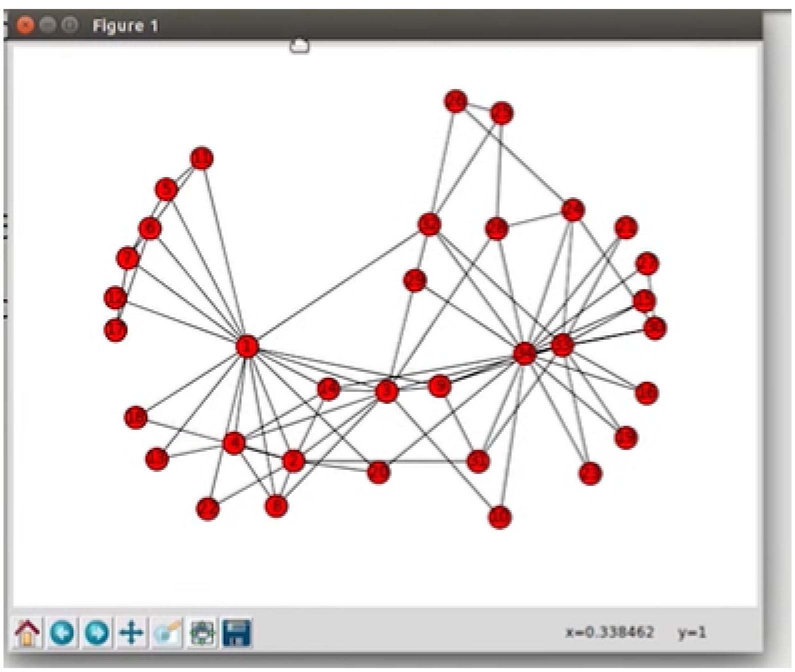
```
# G = nx.read_edgelist('Data_sets/facebook_combined.txt')
G = nx.read_pajek('Data_sets/football.net')
```

```
anamika@anamika-Inspiron-5423:~/Desktop$ python data_sets.py
Name:
Type: MultiDiGraph
Number of nodes: 35
Number of edges: 118
Average in degree:   3.3714
Average out degree:   3.3714
35
118
True
```

```
# G = nx.read_edgelist('Data_sets/facebook_combined.txt')
# G = nx.read_pajek('Data_sets/football.net')

# G = nx.read_pajek('Data_sets/karate.paj')
# G = nx.read_graphml('Data_sets/wikipedia.graphml')
G = nx.read_gexf('Data_sets/EuroSiS Generale Pays.gexf')
```

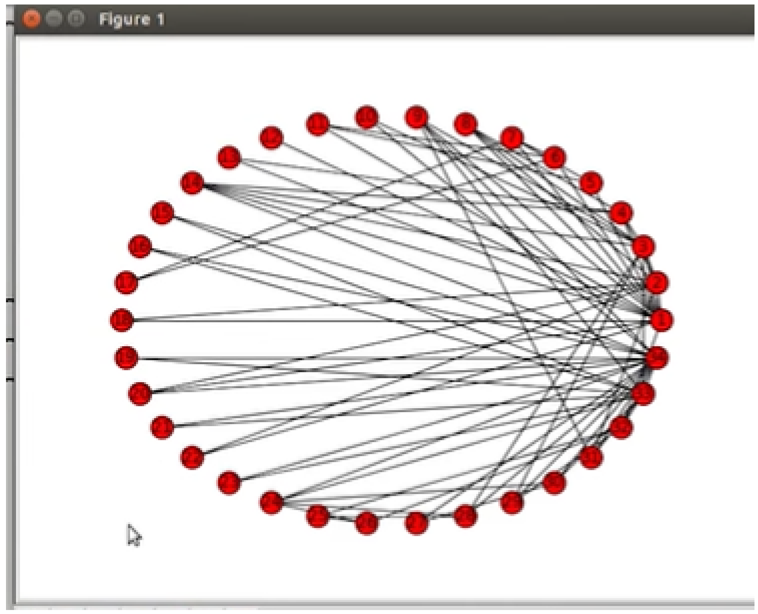
```
G = nx.read_gml('Data_sets/karate.gml')
```

```
nx.draw(G)
plt.show()
```

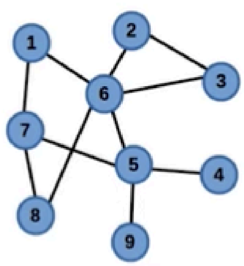


U can zoom and move the graph

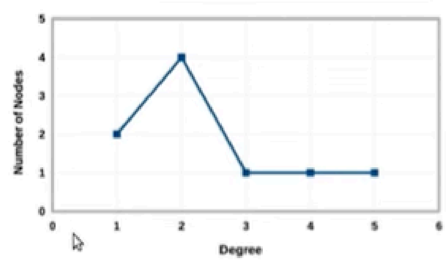
```
# nx.draw(G)
nx.draw_circular(G)
plt.show()
```



Degree Distribution



Degree	Number of Nodes
1	2
2	4
3	1
4	1
5	1




```
In [3]: G = nx.read_gml('Data_sets/karate.gml')

In [4]: nx.degree(G)
Out[4]:
{1: 16,
 2: 9,
 3: 10,
 4: 6,
 5: 3,
 6: 4,
 7: 4,
```

Gives the dictionary showing degree and no. of nodes with that degree

```
In [5]: nx.degree(G).values()
Out[5]:
[16,
 9,
 10,
 6,
 3,
 4,
 4,
 4,
```

gives all possible degree values

```
In [6]: set(nx.degree(G).values())
Out[6]: {1, 2, 3, 4, 5, 6, 9, 10, 12, 16, 17}
```

Makes a set and removes all duplicate values

```
In [7]: list(set(nx.degree(G).values()))
Out[7]: [1, 2, 3, 4, 5, 6, 9, 10, 12, 16, 17]
```

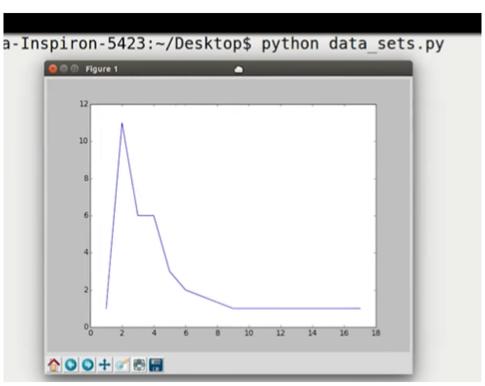
—--------------- in main fucntn —---------

Making degree plot -

```python
import networkx as nx
import matplotlib.pyplot as plt

def plot_deg_dist(G):
    all_degrees = nx.degree(G).values() #all the degrees
    unique_degrees = list(set(all_degrees)) #all the unique degrees

    count_of_degrees = []

    for i in unique_degrees:
        x = all_degrees.count(i)
        count_of_degrees.append(x)

    plt.plot(unique_degrees, count_of_degrees)
    plt.show()
```

```python
G = nx.read_gml('Data_sets/karate.gml')
```

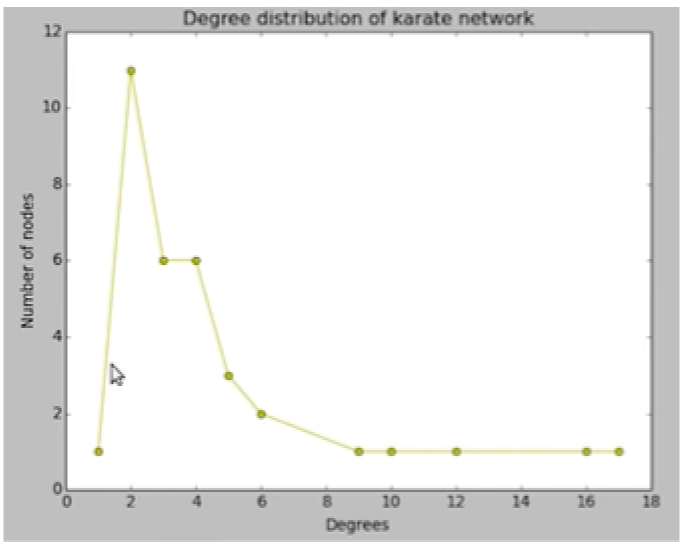
```python
plot_deg_dist(G)
```



.

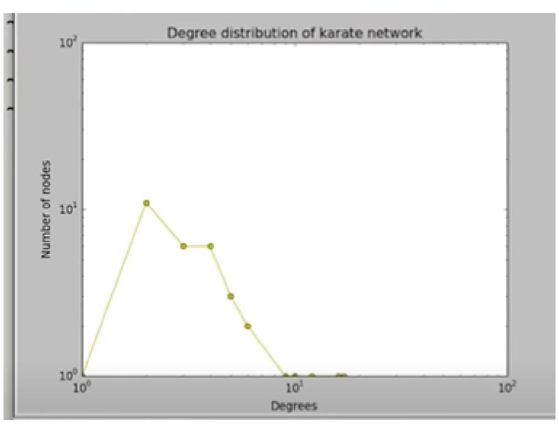
Adding x y labels

```python
def plot_deg_dist(G):
    all_degrees = nx.degree(G).values() #all the degrees
    unique_degrees = list(set(all_degrees)) #all the unique degrees

    count_of_degrees = []

    for i in unique_degrees:
        x = all_degrees.count(i)
        count_of_degrees.append(x)

    plt.plot(unique_degrees, count_of_degrees, 'yo-')
    plt.xlabel('Degrees')
    plt.ylabel('Number of nodes')
    plt.title('Degree distribution of karate network')
    plt.show()
```

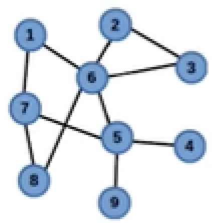


We can make log log graph to check if this follows powerlaw

```python
plt.loglog(unique_degrees, count_of_degrees, 'yo-')
```



Density



$$\text{Density} = \frac{\text{Number of edges present}}{\text{Total possible edges}}$$

$$\text{Density(G)} = \frac{11}{36} = 0.31$$

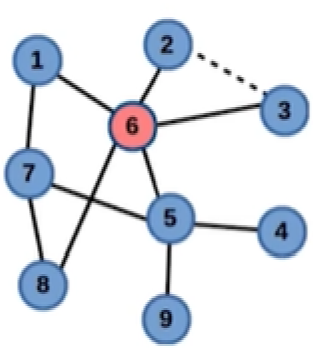
```
In [9]: G = nx.complete_graph(100)

In [10]: nx.density(G)
Out[10]: 1.0

In [11]: H = nx.Graph()

In [12]: H.add_nodes_from([1,2,3,4])

In [13]: nx.density(H)
Out[13]: 0.0
```

```
print 'Density is', nx.density(G)
```

```
anamika@anamika-Inspiron-5423:~/Desktop$ pytho
Density is 0.139037433155
```

Clustering Coefficient

$$\text{Clustering Coefficient} = \frac{\text{Actual Number of friendships}}{\text{Total possible friendships}}$$



Clustering coeff (6) see its neighbours - 1 2 3 5 8
Now total can be 10 links but only 1 link is present between these neighbours

$$\text{Clustering Coefficient(6)} = \frac{1}{10}$$

```
nx.clustering(G)
```
gives dictionary giving clustering coeff value of every node

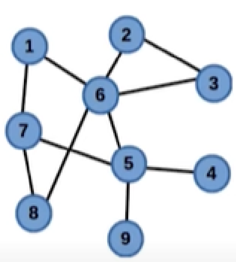
```
for i in nx.clustering(G).items():
    print i
```

```
anamika@anamika-Inspiron-5423:~/Desktop$ python data_sets
(1, 0.15)
(2, 0.3333333333333333)
(3, 0.24444444444444444)
(4, 0.6666666666666666)
(5, 0.6666666666666666)
(6, 0.5)
(7, 0.5)
(8, 1.0)
(9, 0.5)
```

```
print nx.average_clustering(G)
```
tell avg clustering value

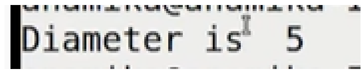
```
(34, 0.11029411
0.570638478208
anamika@anamika
```

Diameter

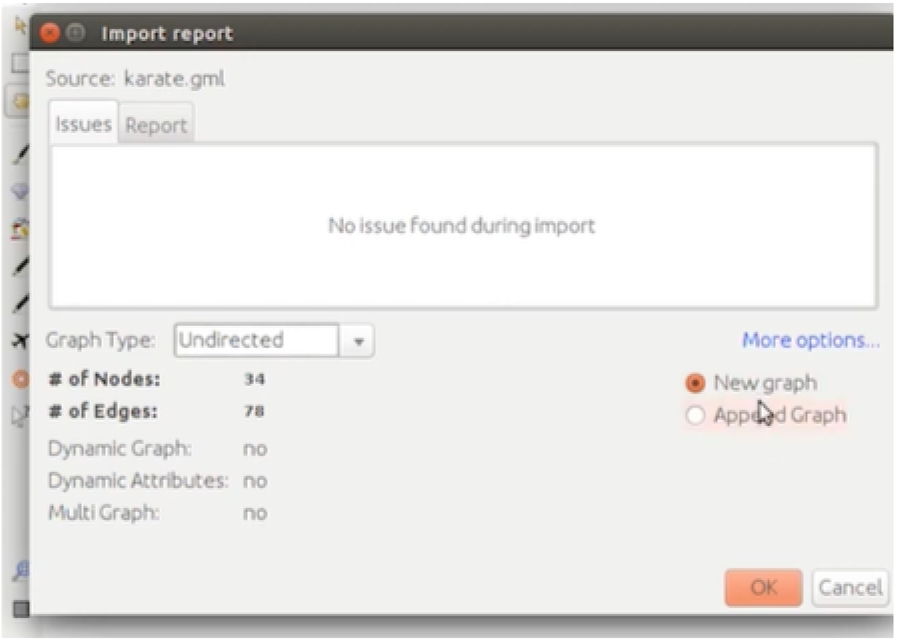


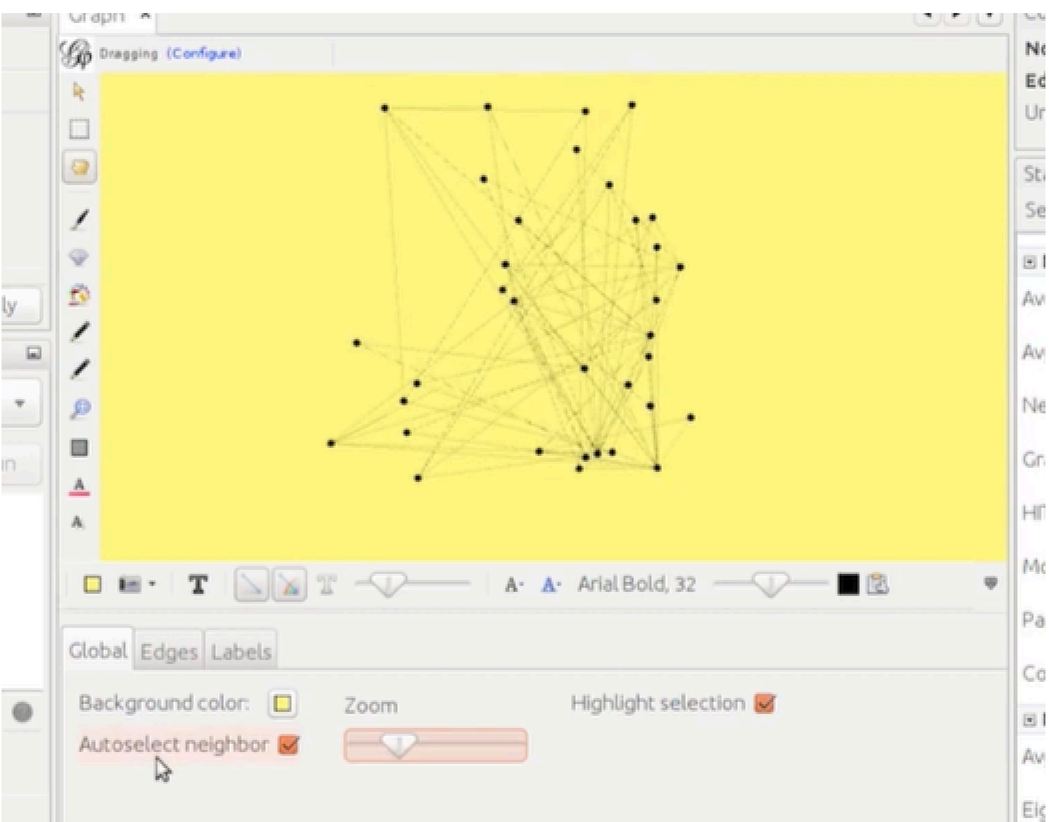
Diameter = 3

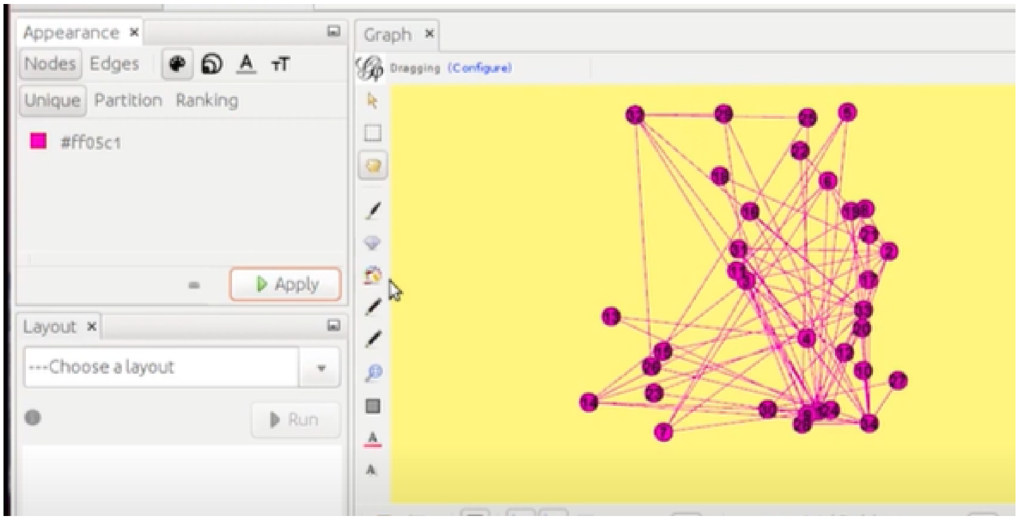
so what is the diameter of a networkdiameter
is basically the maximum shortest path that
we have to travel to go from one node to the
otherfor exampleif you if you know about all
pair shortest path algorithm it basically
the returns the metrics wherethe the values
are the length of the shortest path being the
two nodesso is it as that for everyevery pair
of thenodes so whatever is the maximum value in
that metrics will be the diameter of the network
in other words its the shortest path between
two most distant nodes in the network sofor



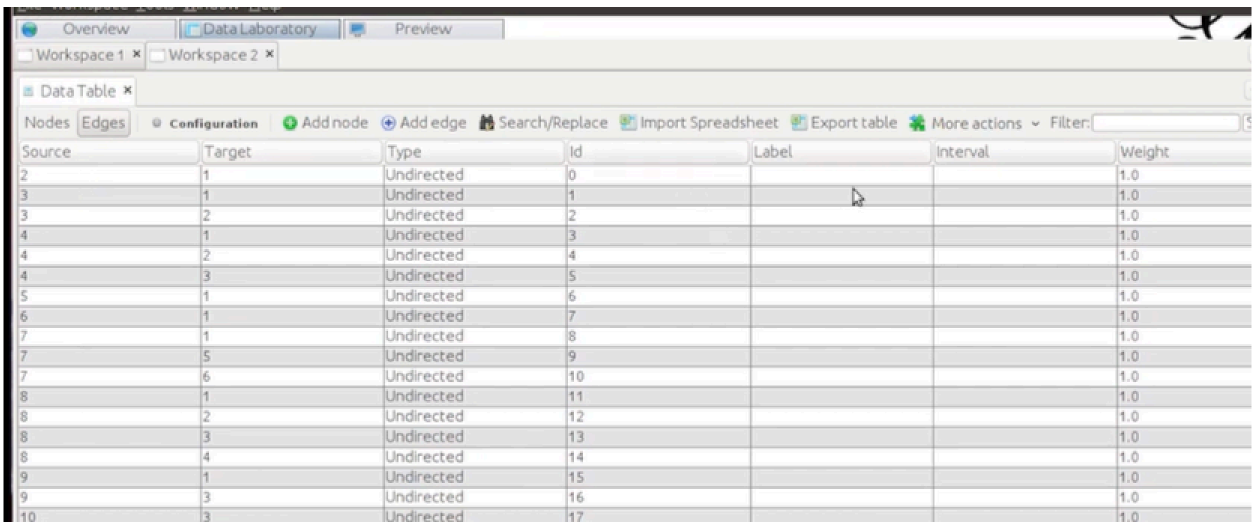
Lecture 22 - Datasets: Analysing Using Gephi





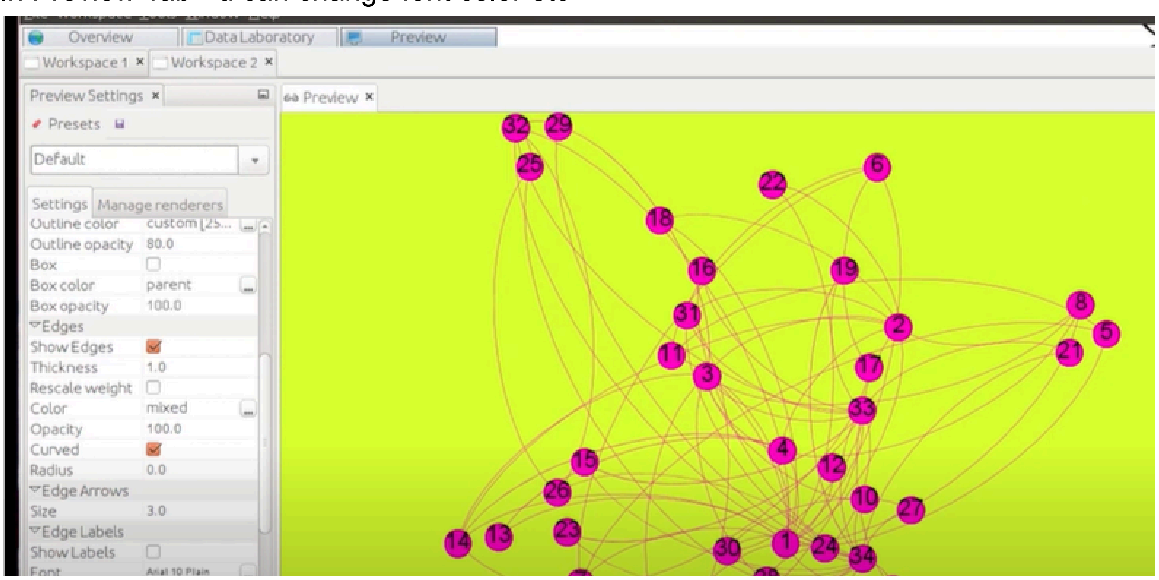


Data laboratory has the info in a table

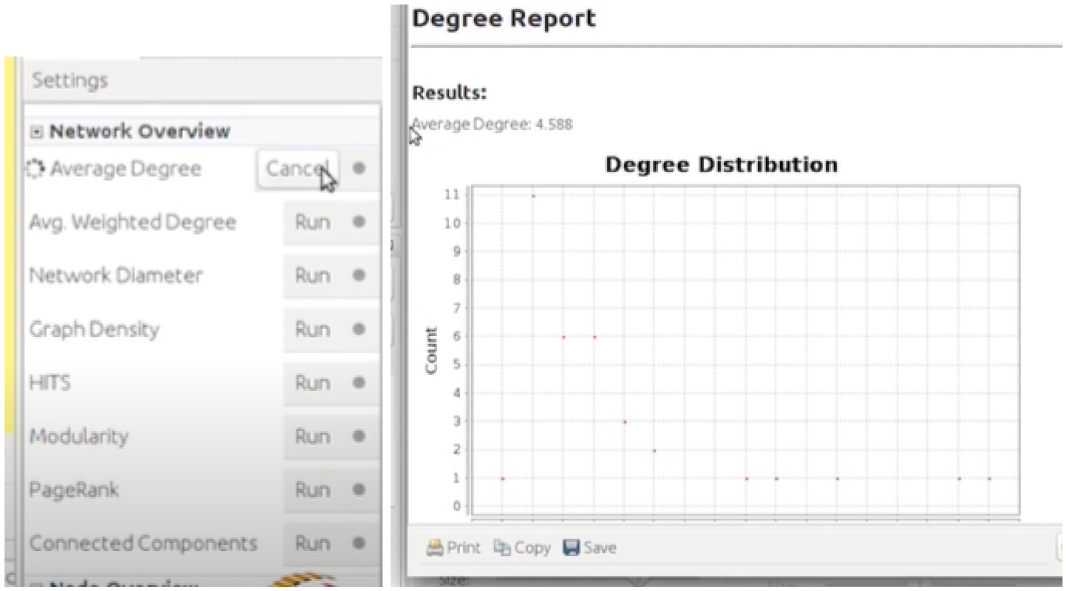


Source	Target	Type	Id	Label	Interval	Weight
2	1	Undirected	0			1.0
3	1	Undirected	1			1.0
3	2	Undirected	2			1.0
4	1	Undirected	3			1.0
4	2	Undirected	4			1.0
4	3	Undirected	5			1.0
5	1	Undirected	6			1.0
6	1	Undirected	7			1.0
7	1	Undirected	8			1.0
7	5	Undirected	9			1.0
7	6	Undirected	10			1.0
8	1	Undirected	11			1.0
8	2	Undirected	12			1.0
8	3	Undirected	13			1.0
8	4	Undirected	14			1.0
9	1	Undirected	15			1.0
9	3	Undirected	16			1.0
10	3	Undirected	17			1.0

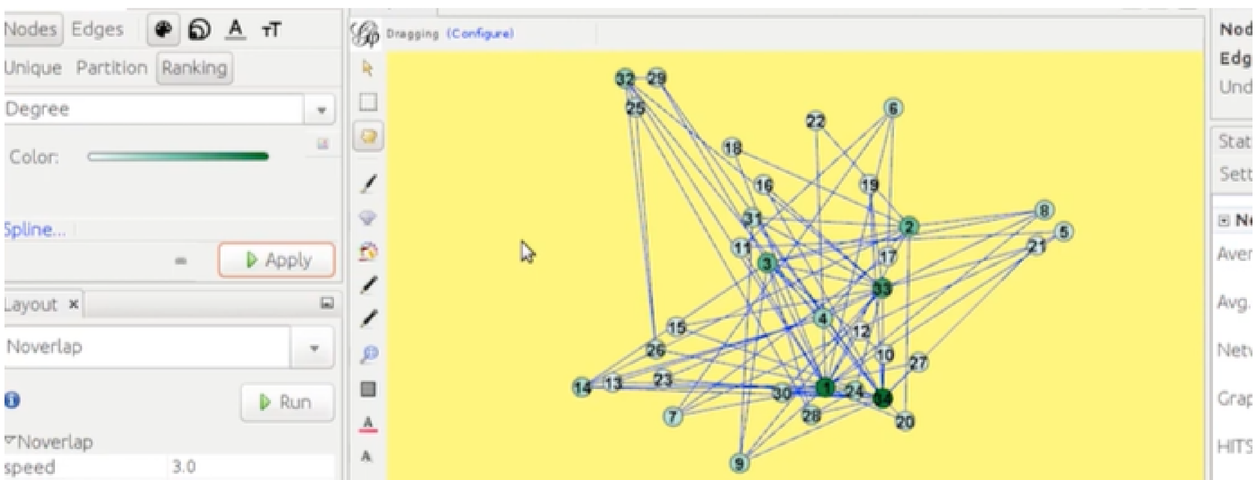
In Preview Tab - u can change font color etc



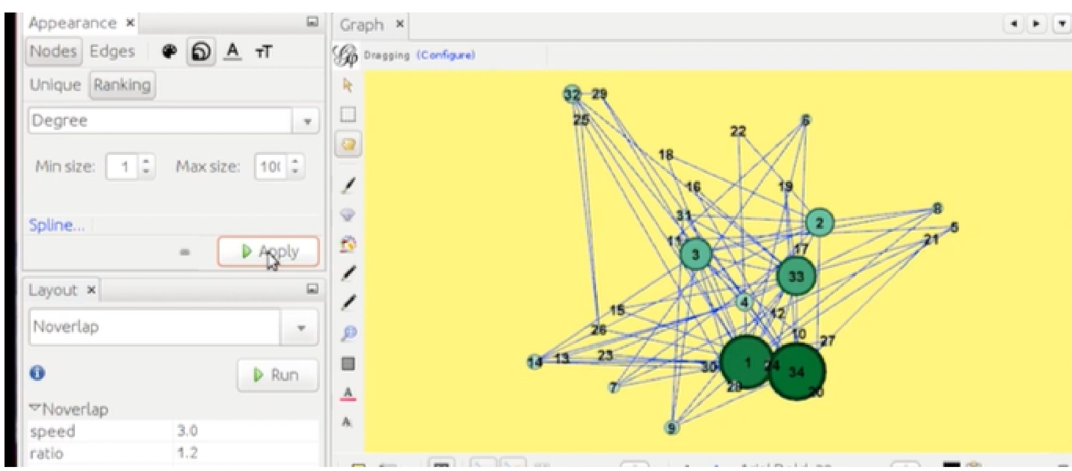
In overview =



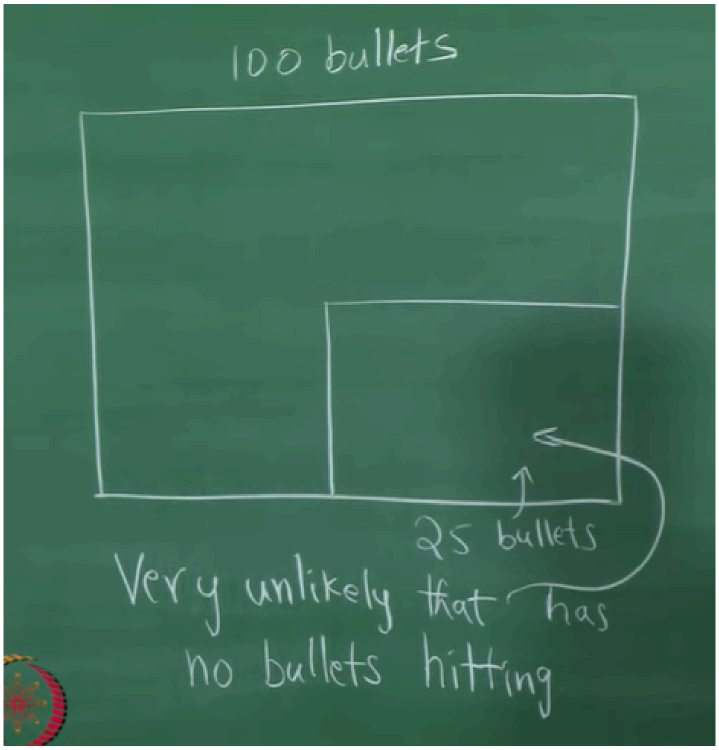
Degree ranking in terms of colro



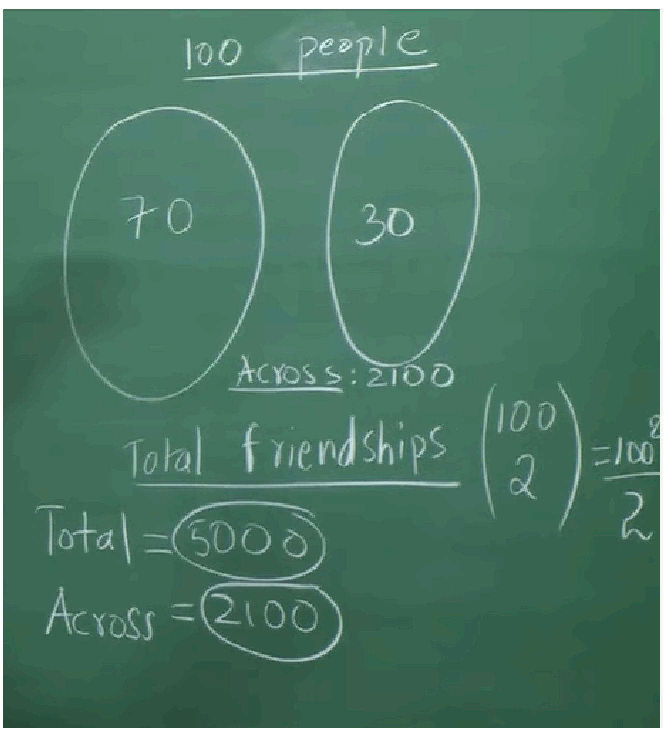
In size -



Lecture 23 - Introduction : Emergence of Connectedness
MAIN QUES - WHEN EXACTLY WE REACH A CONNECTED GRAPH ?

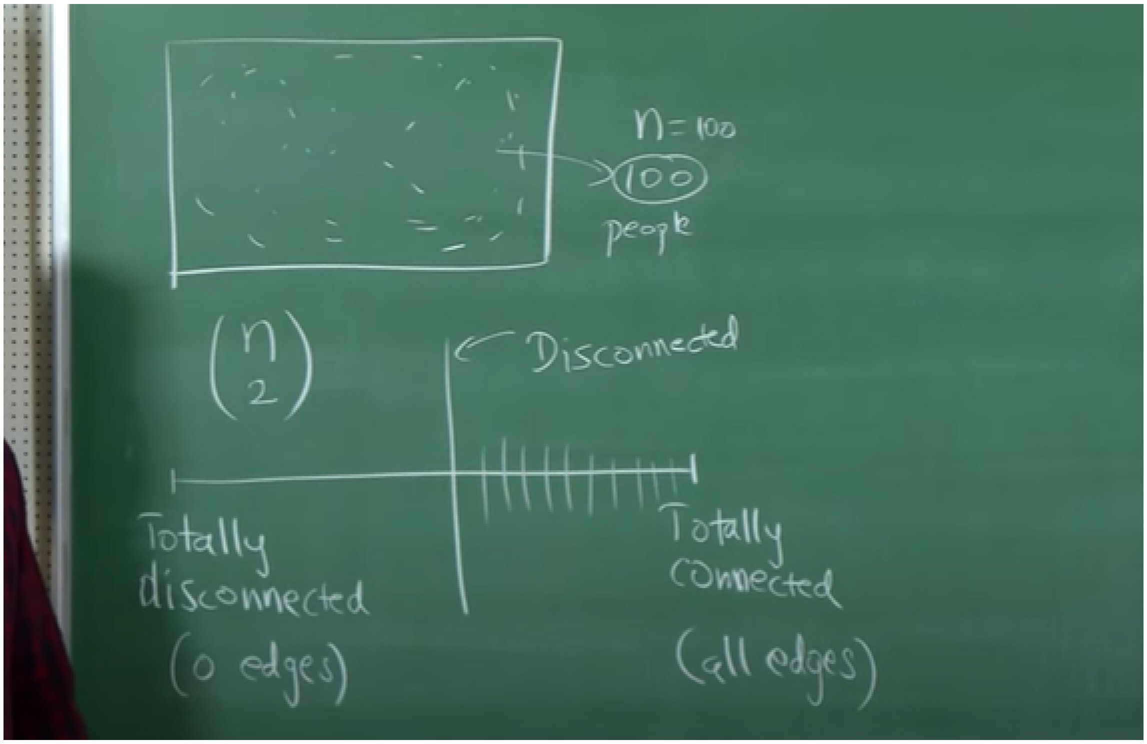


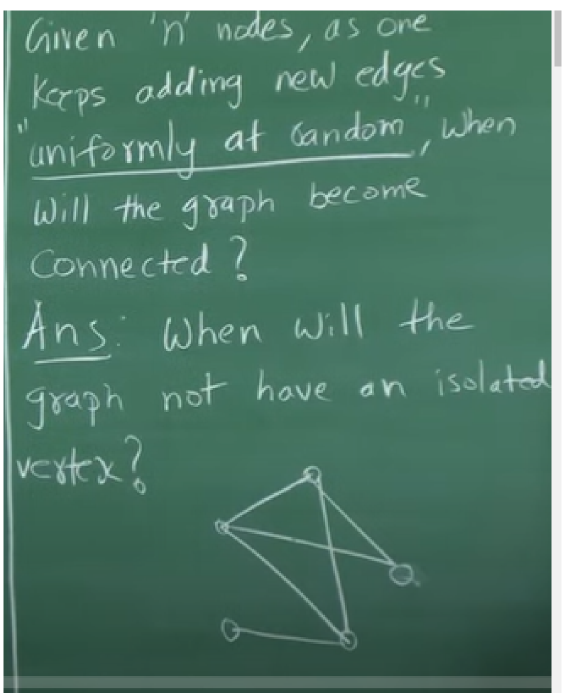
LIKEWISE ,



THERE COuld have been 5000 across friendships but 2100 happened only (70*30)
But having 0 across is very very unlikely => there should be a few friendships across

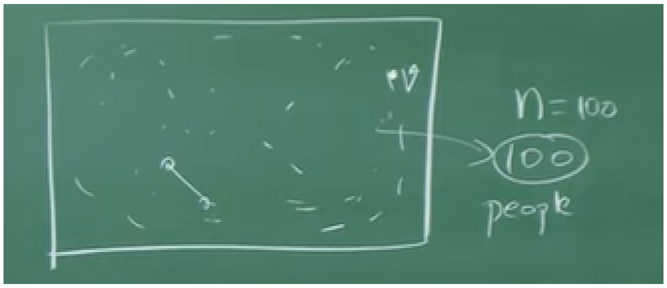
Lecture 24 - Advanced Material : Emergence of Connectedness
You have 100 people, put 100C2 edges for all. Keep on removing one one edge from the graph. There comes a point when a=the graph becomes disconnected. WHEN?





,

Prob that v is not isolated after putting one edge- (SOME MISTAKE IN LECTURE)



Prob(v is isolated) when an edge is included

$= \frac{2}{n}$

$\left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$

Prob(v is isolated) when an edge is included

$= \frac{2}{n}$

Prob(v not isolated) $= \left(1 - \frac{2}{n}\right)$

Prob(v not isolated after k edges being put) $= \left(1 - \frac{2}{n}\right)^k$

$k=1$ Pr(v not isolated) $= 1 - \frac{2}{n}$

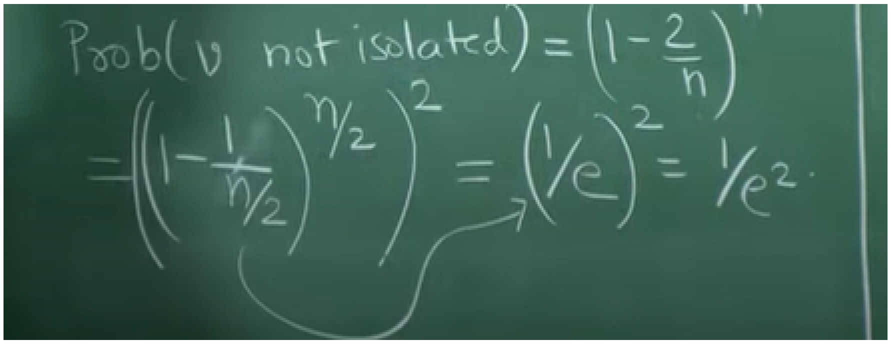
$k=2$ . . . . $= \left(1 - \frac{2}{n}\right)^2$

$k=n$ . . . . $\left(1 - \frac{2}{n}\right)^n$

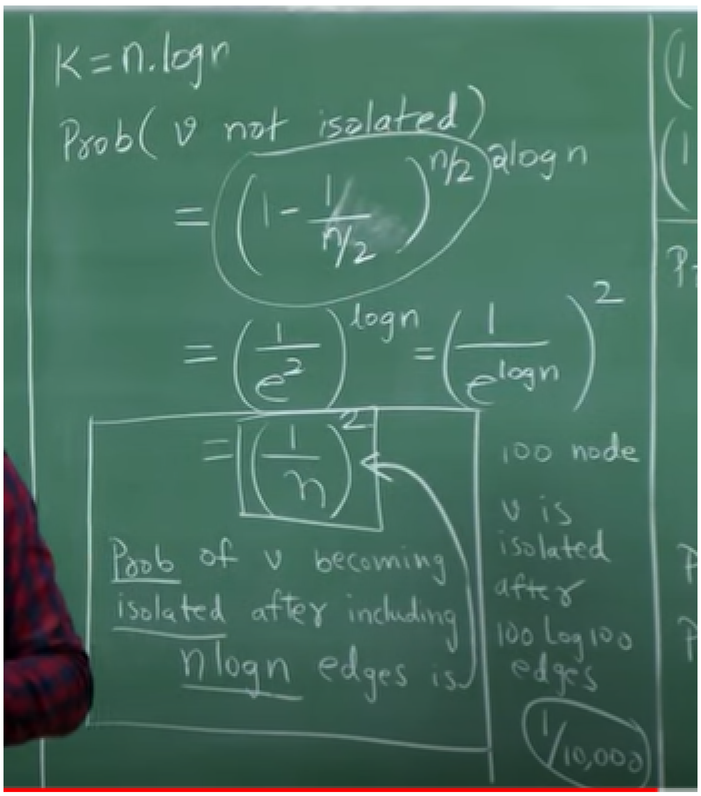
When we include 'n' edges,

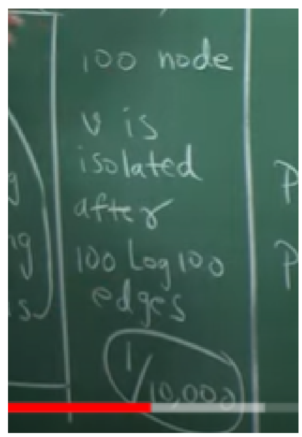
Prob(v not isolated) $= \left(1 - \frac{2}{n}\right)^n$

$= \left(1 - \frac{1}{n/2}\right)^n$



Puting K = n.logn





IN A 100node graph, v is isolated after 100 log 100 edges and this has a prob of 1/10000

CONCLUSION – IF YOU PUT N.LOGN EDGES THE GRAPH BECOMES CONNECTED

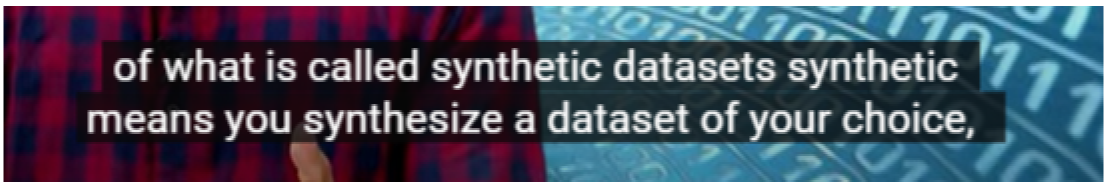
Lecture 25 -Programming Illustration : Emergence of Connectedness

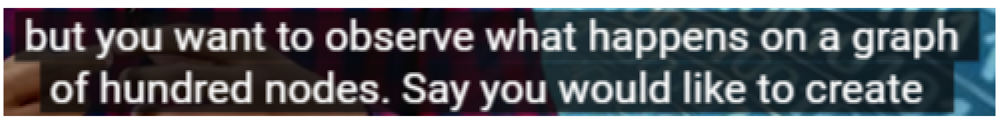
CODE IN VIDEO

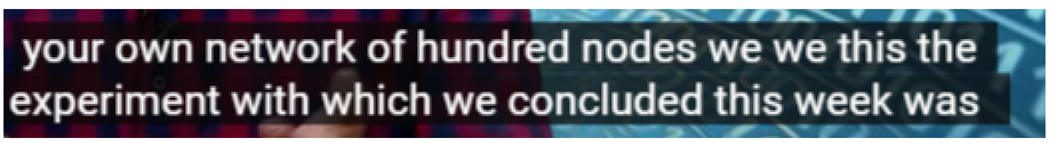
Lecture 26 - Summary to Datasets

Gephi usage
Coding for analysis
Emergence of connectedness
Graph becomes connected when nlogn edges
Code also proved this

Power of Synthestic Datasets -







8) Given is a graph G with $|V| = n$ number of nodes and $|E|$ number of edges. In which of the following cases, we can guarantee that G is connected? ***1 point***

- $|E| = n$
- $|E| = n - 1$
- $|E| = n(n-1)/2$
- $|E| = n\ \log_2 n$

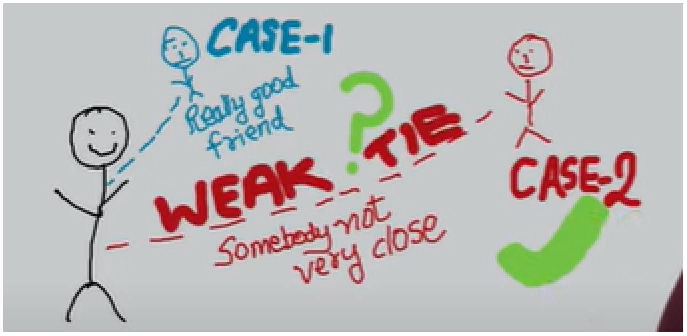
No, the answer is incorrect.
Score: 0

Accepted Answers:
$|E| = n(n-1)/2$

WEEK 3 - STRONG AND WEAK RELATIONSHIPS

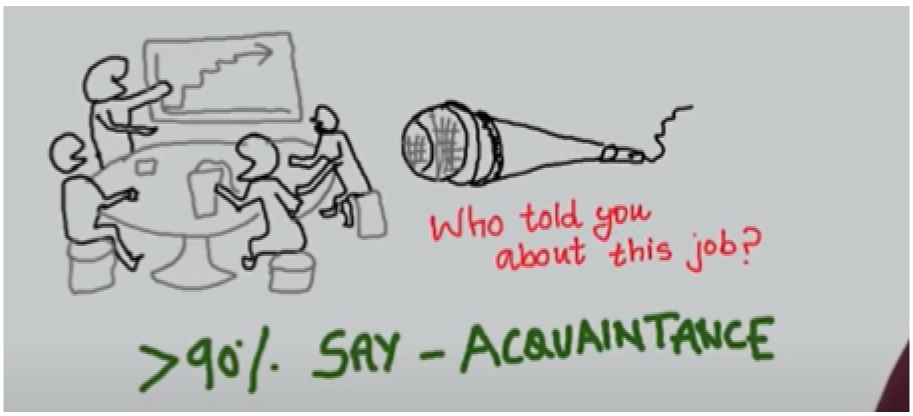
Lecture 27 - Introduction

JOB OPPORTUNITY BY TWO TYPES OF PEOPLE -



most cases are by unknown people

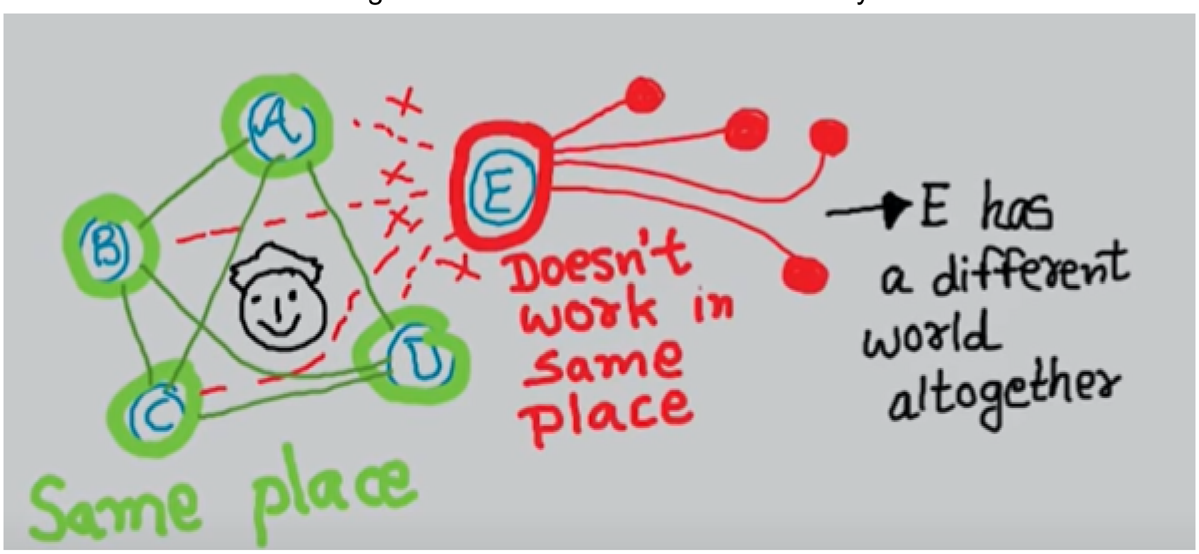
Lecture 28 - Granovetter's Strength of weak ties



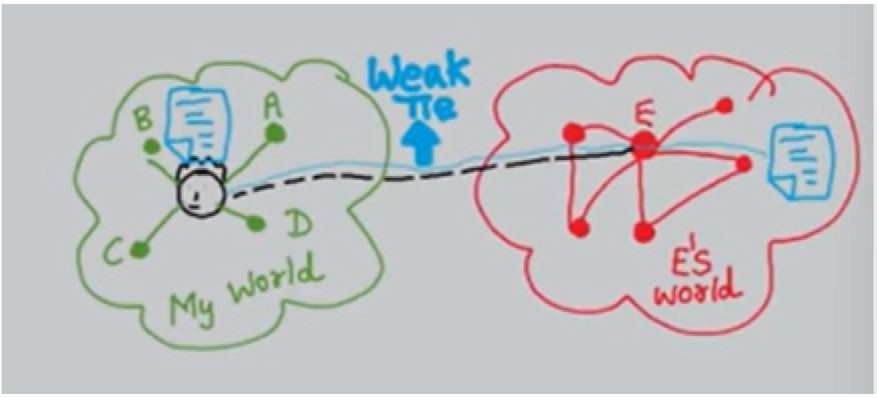
rarely close people tell u about job opportunity

WHY??

A B C D and me work wrk together and E from different worl dis my frnd

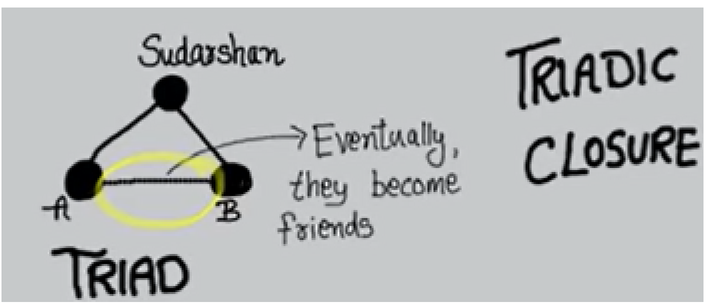


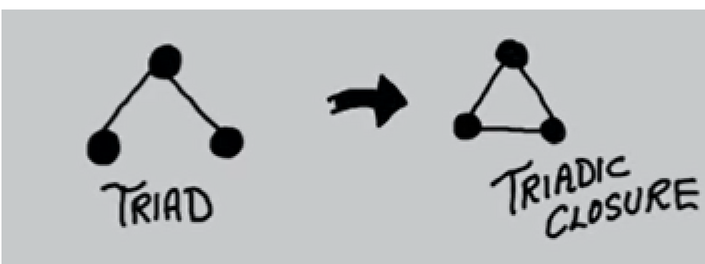
E can give me new info

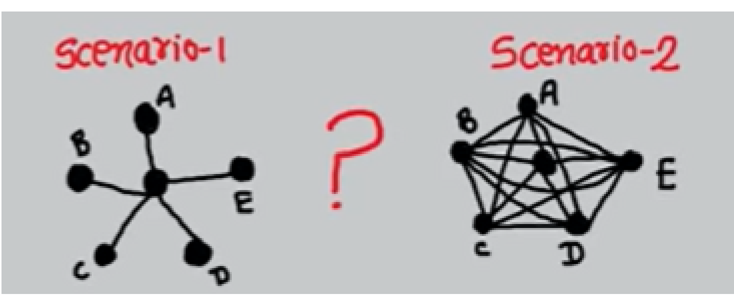


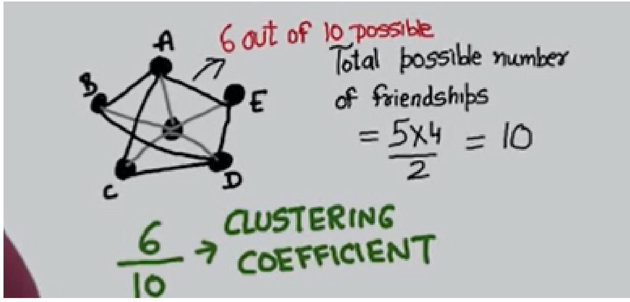
Granovetter's conducted expt and told that weak ties help u in job related info

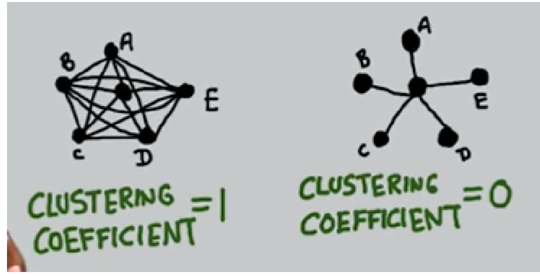
Lecture 29 - Triads, clustering coefficient and neighborhood overlap



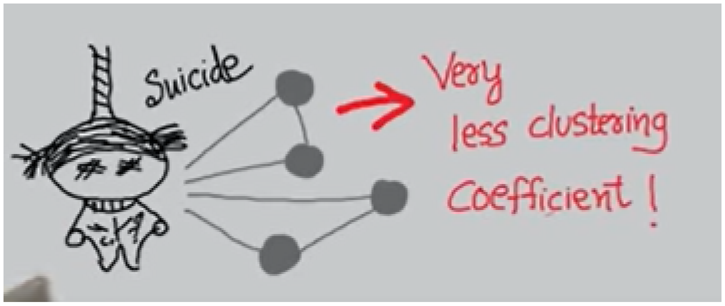




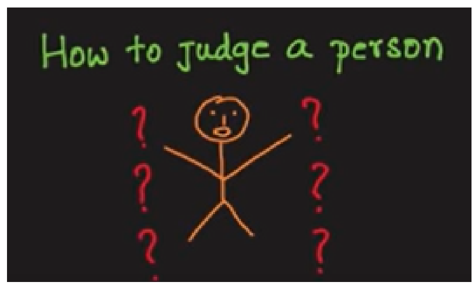


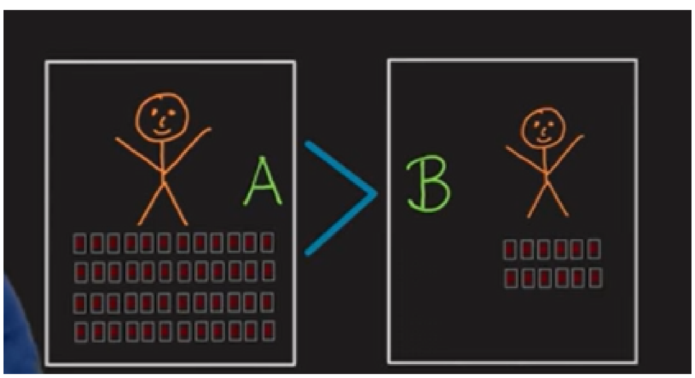


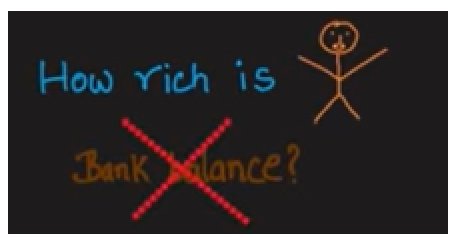
Tells about the strength of the friendship



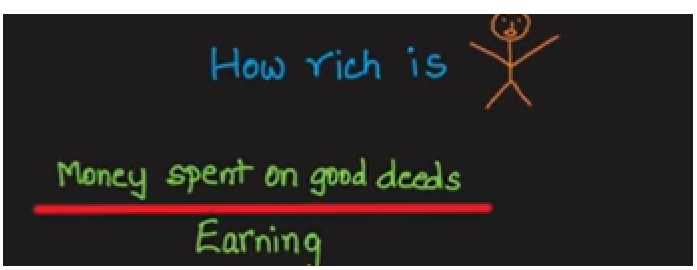
—--.

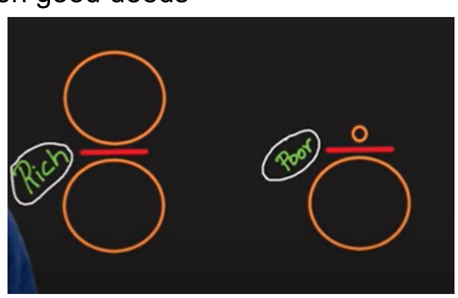


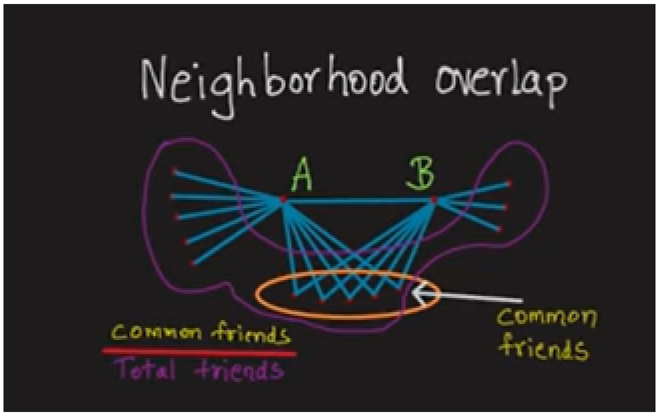




Redefining richness by checking the proportion spent on good deeds

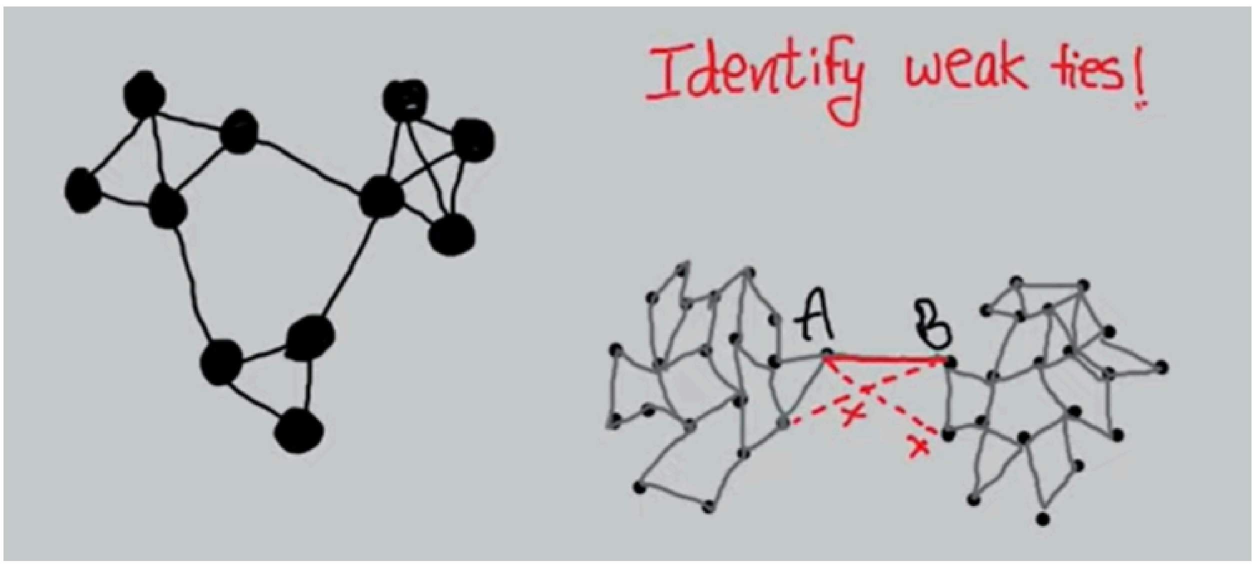




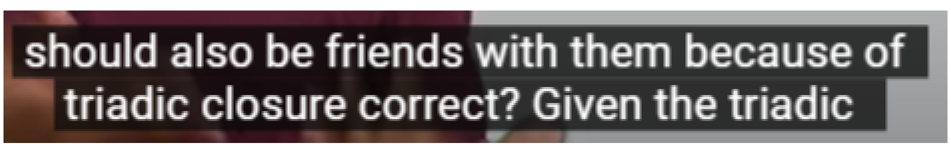


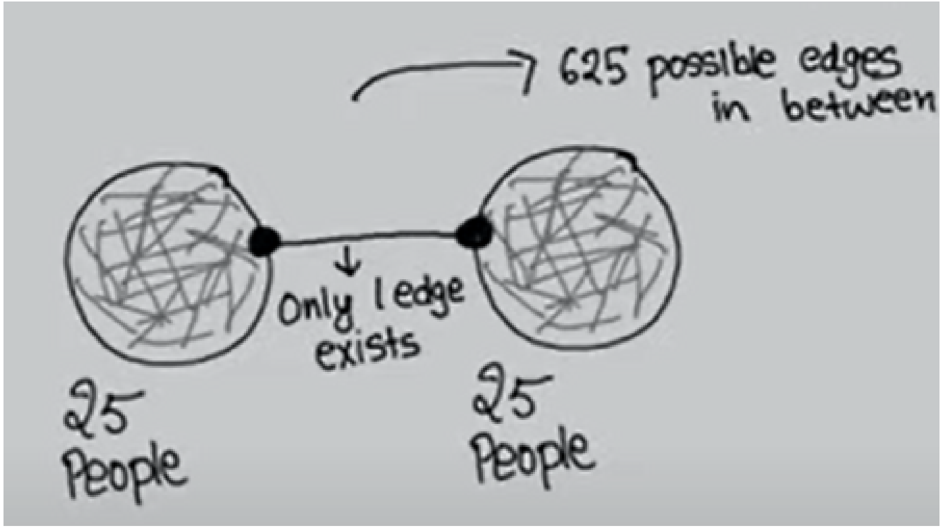
friends of A & B / total friends (of both A &B)

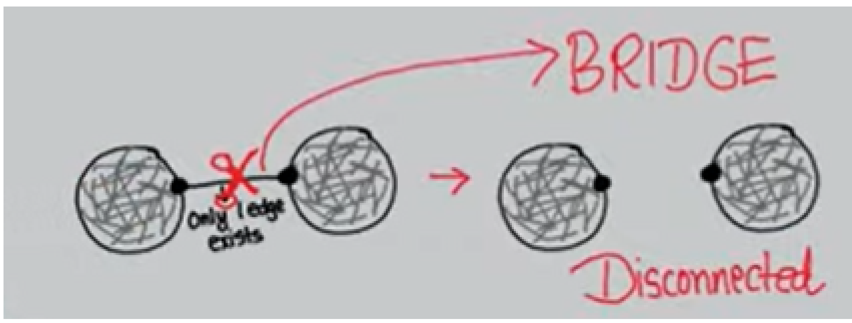
Lecture 30 - Structure of weak ties, bridges, and local bridges



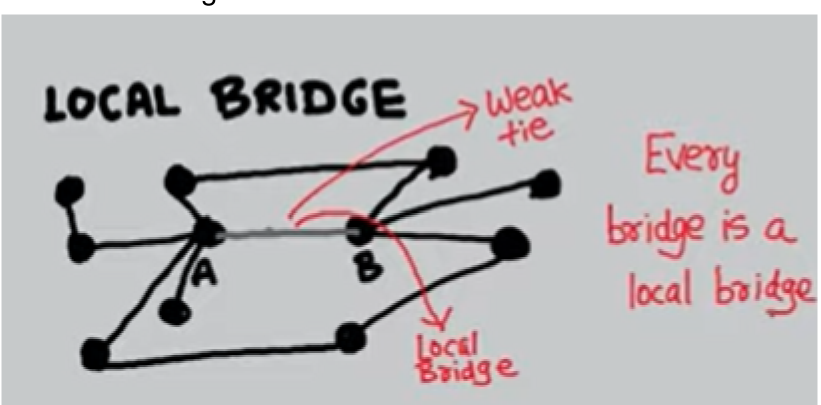
Red ones are a weak tie

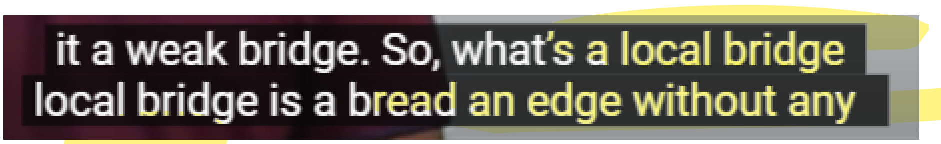


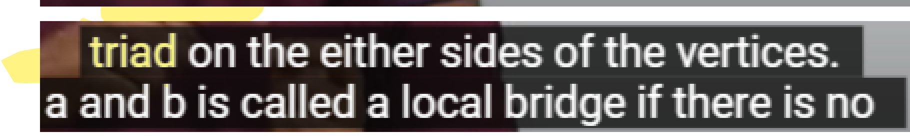




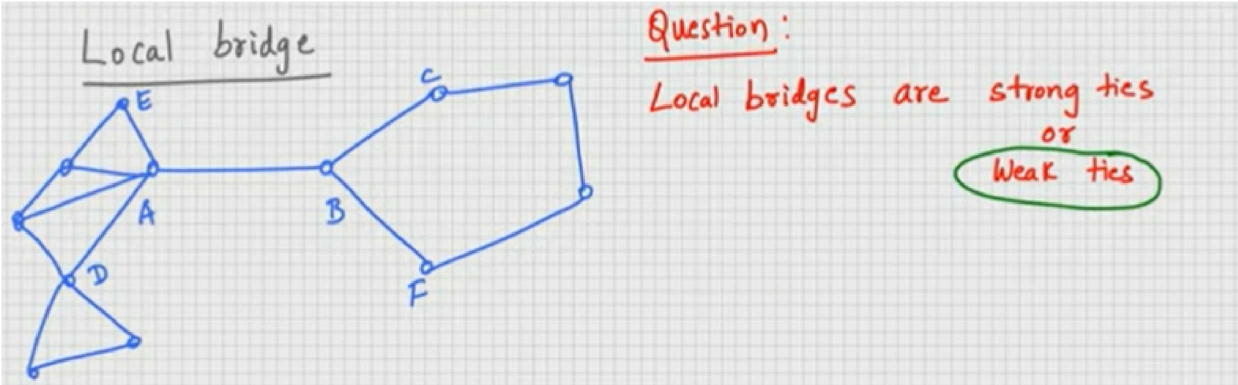
graph gets disconnected if u remove the bridge



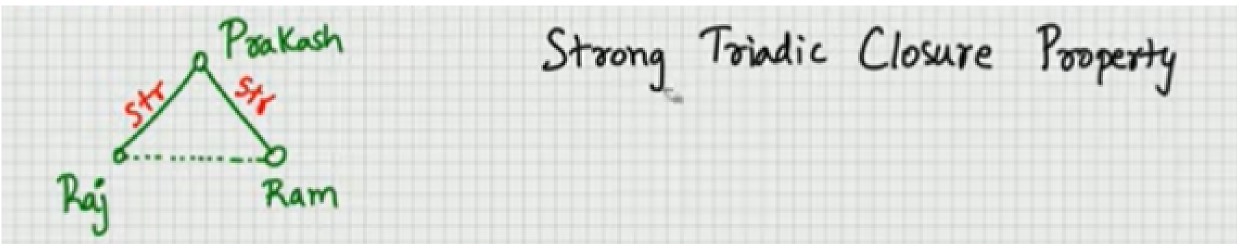


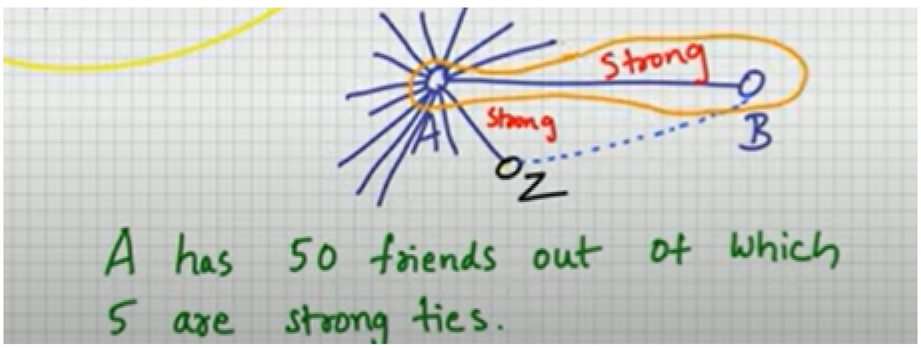


LOCAL BRIDGE == WEAK TIE (always)

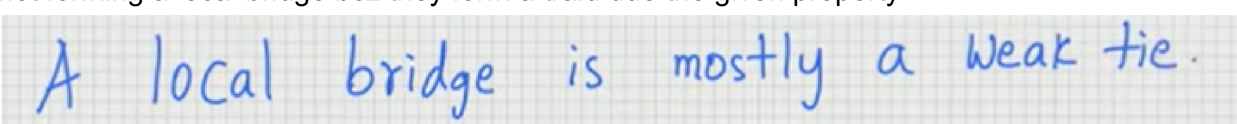


IF PRAKASH has very very strong tie with 2 friends, they also tend to know each other



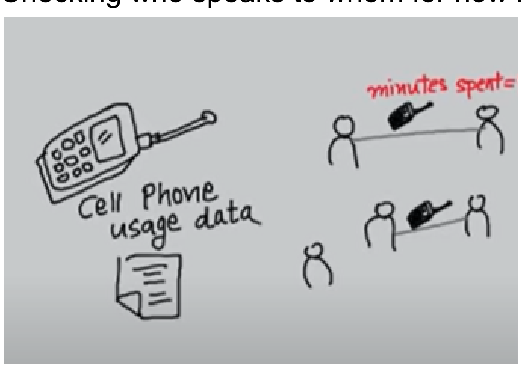


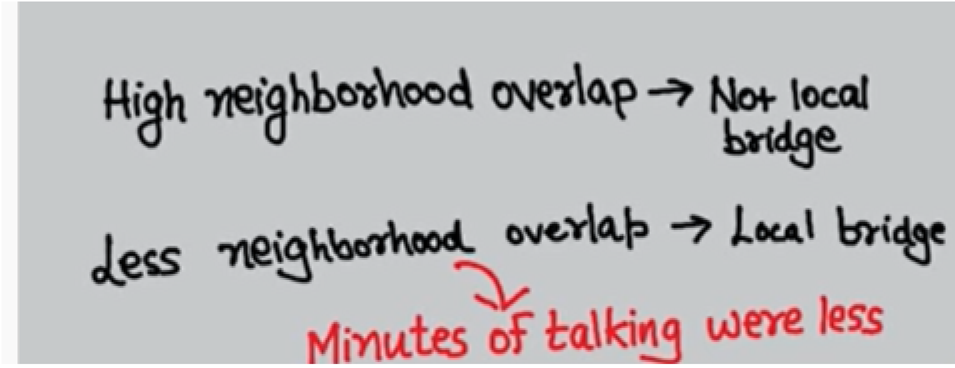
in this case, A and B are not forming a local bridge bcz they form a traid due the given property

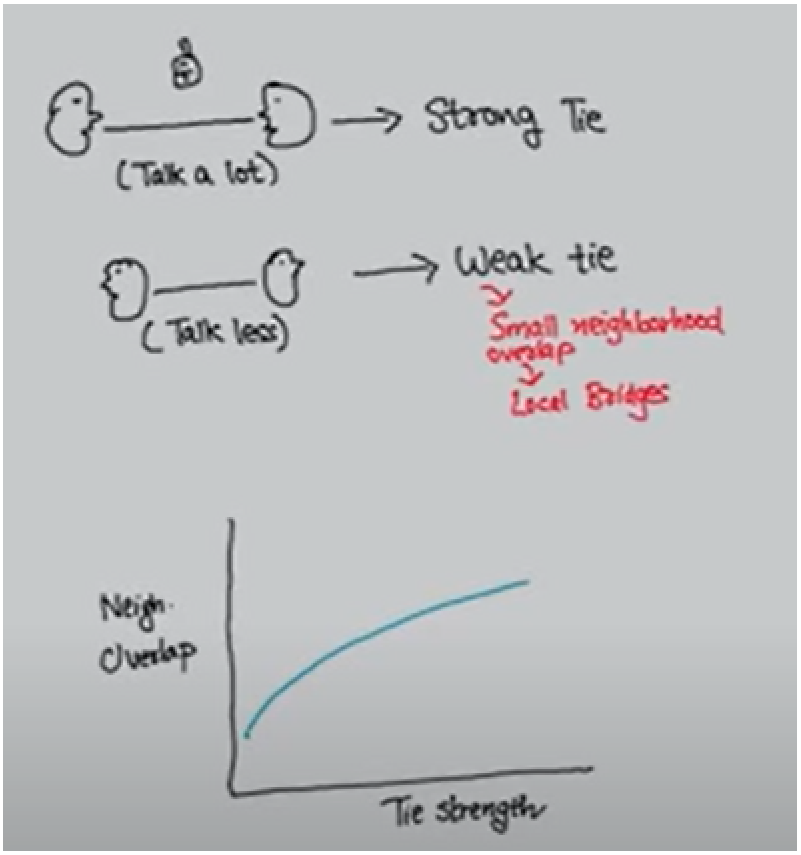


Lecture 31 - Validation of Granovetter's experiment using cell phone data

Checking who speaks to whom for how many mins - fo 4.5mnths

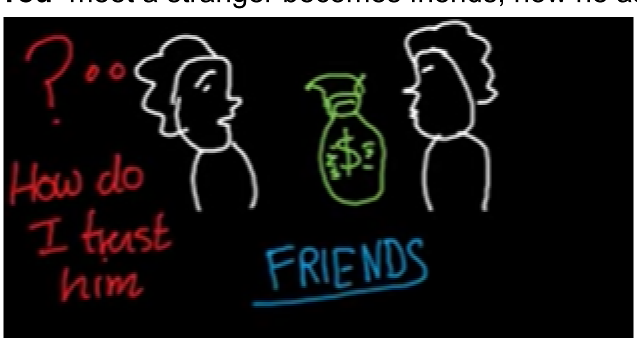




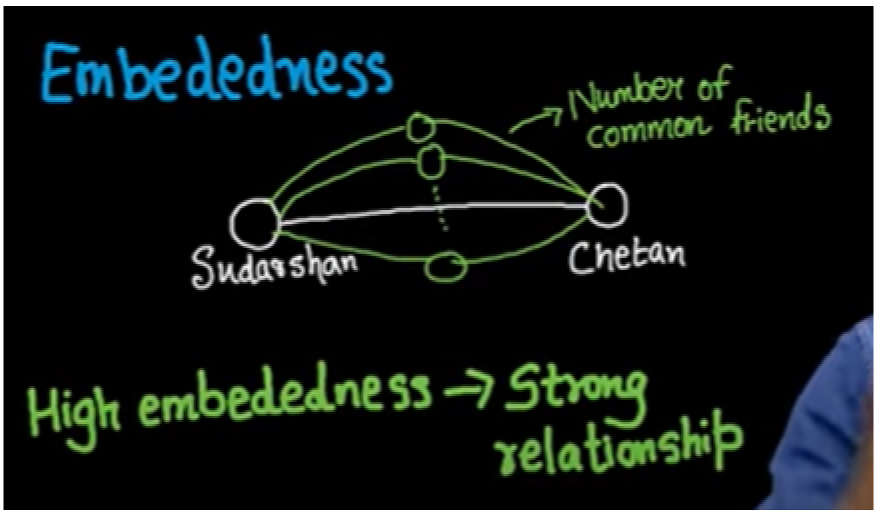


Lecture 32 - Embeddedness

You meet a stranger becomes friends, now he asks u money



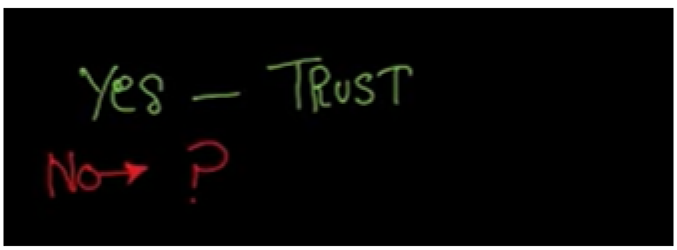
But if good frnd asks for money



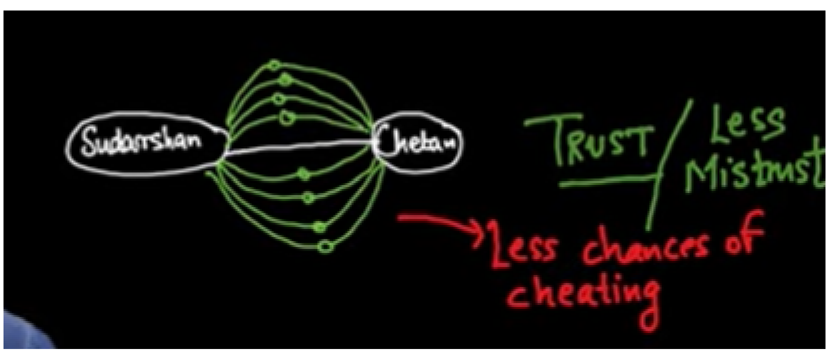
if stranger knows common friends i might trust him

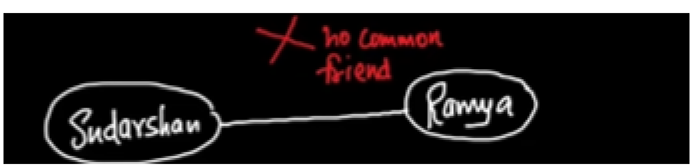
Banker looks for guarantee

SHOULD YOU HAVE HIGH EMBEDDEDNESS IN RSHIPS ?? ANS = yes and no both



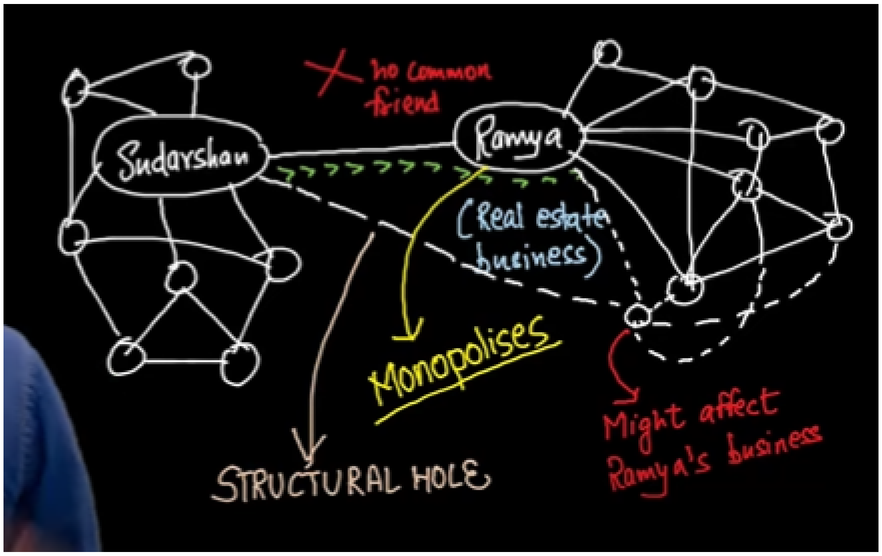
Lecture 33 - Structural Holes





can this be advantageous??

Yes, can help u to know people from different world



People on left have to go throught sudarhsan then ramya then others to contact others

that side right so high embeddedness in the
context of personal friendship adds in trust

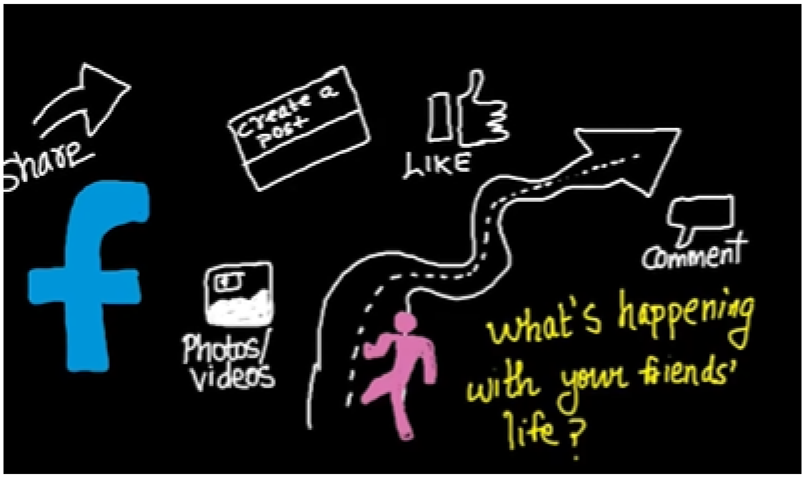
even in the context of business adds in trust
higher the embeddedness more the trust so the

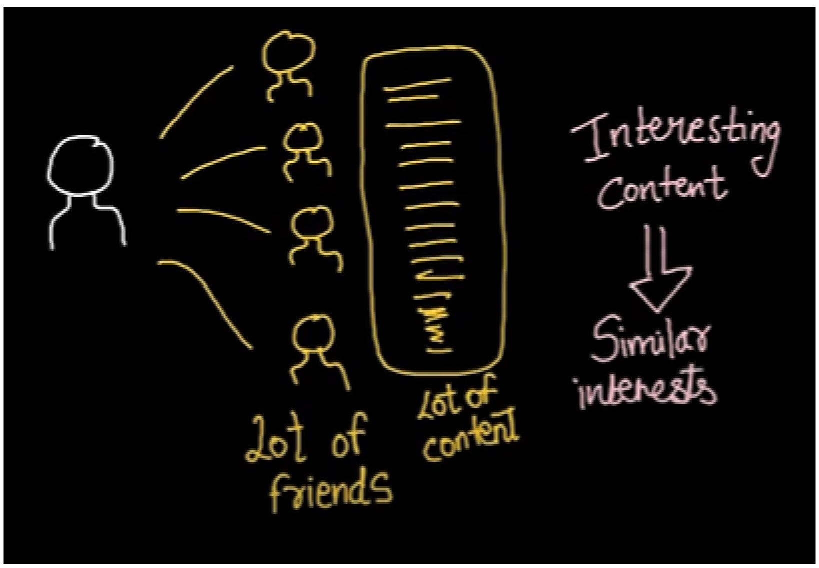
relationship benefits but in such a situation
a huge structural hole exist in the network

structure and whenever there is an edge between
a people like sudarshan and ramya ramya benefits

from this transa any transaction between sudarshan
and her simply because she monopolizes.

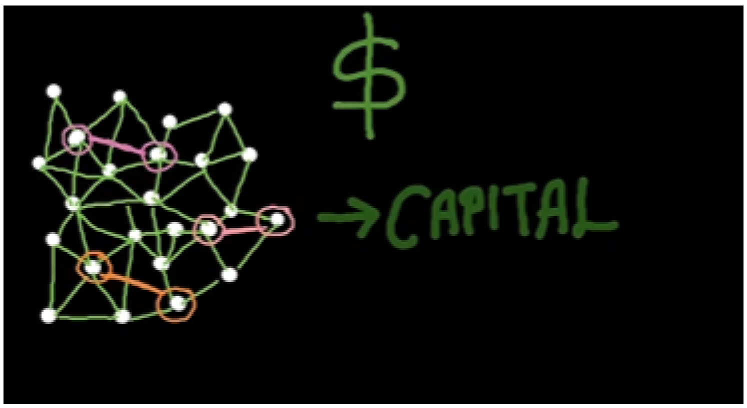
Lecture 34 - Social Capital
If facebook wants to get popular



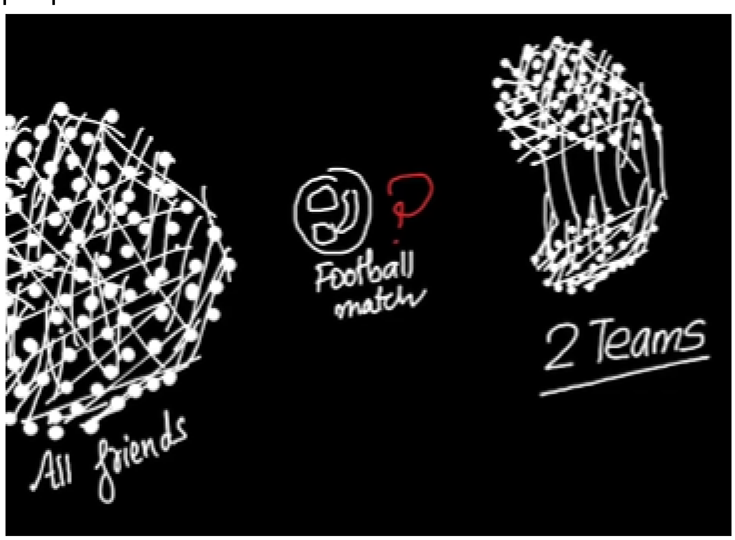


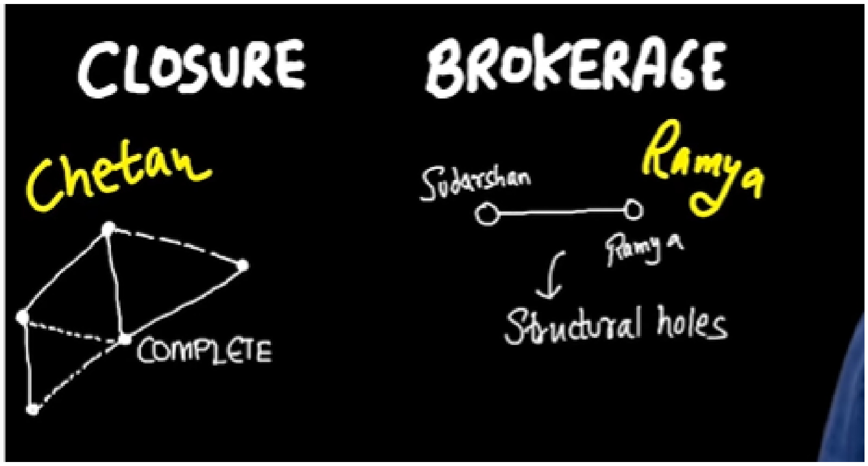
Facebook helps u make new friends , which have similar interest like u

Tries to get u follow interesting friends



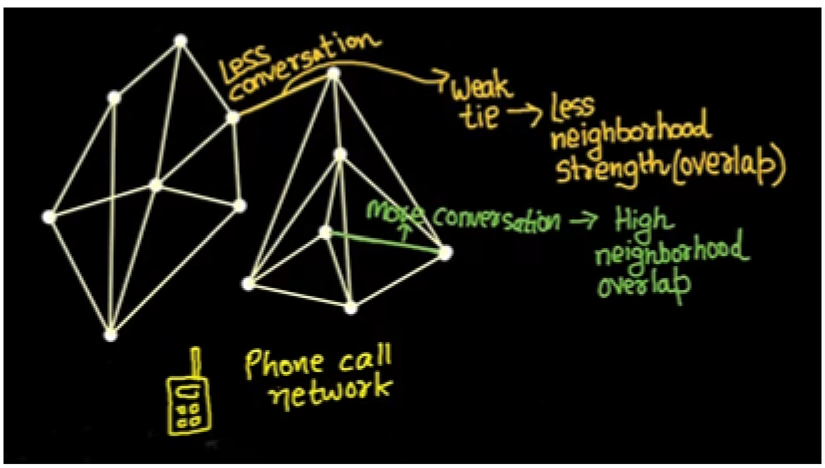
trying to make friends between 5 people who dont like each other



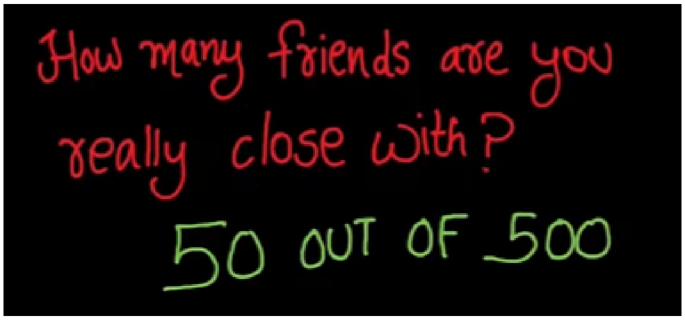


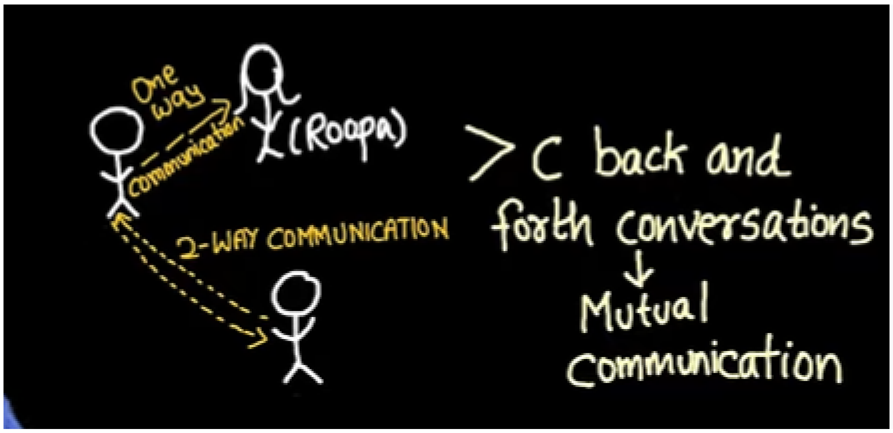
Closure and brokerage both are very imp

Lecture 35 - Tie Strength, Social Media and Passive Engagement

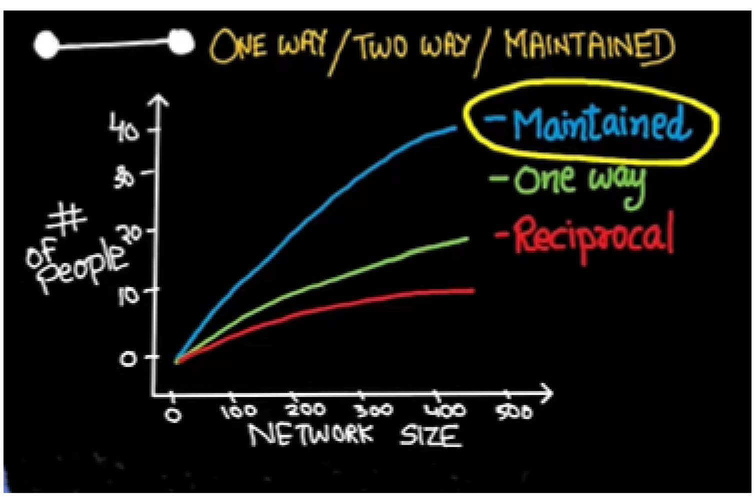


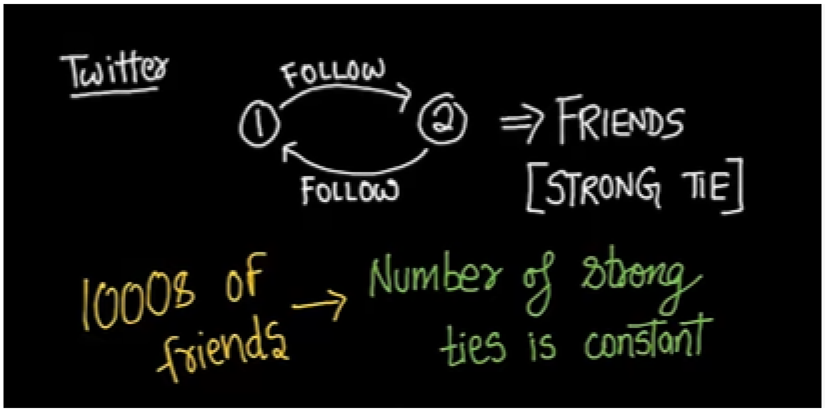
We may not be in touch with a frnd, but he knows whats happening in my life - passive engagement





maintained = passive rships



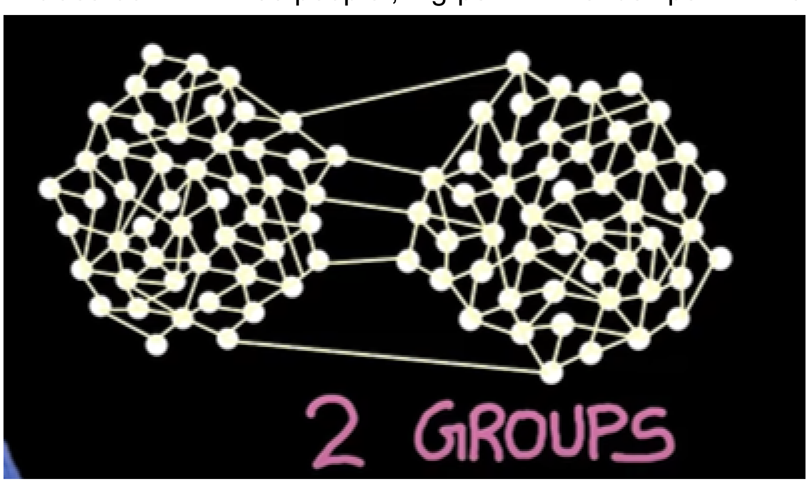


No. of direct msgs shared were remaining constant no matter how many followers u had

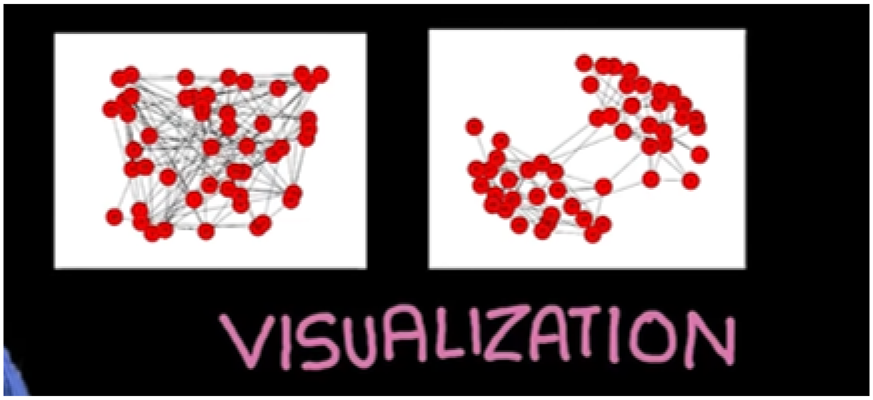
This no. of close or strong tie converges at a certain point

Lecture 36 - Betweenness Measures and Graph Partitioning

A classroom with 100 people , 2 grps with friendships within and less across



After visualisation in other way we realised that thre is grouping in the graph



how do you define a community? A community is a bunch of nodes where the connections

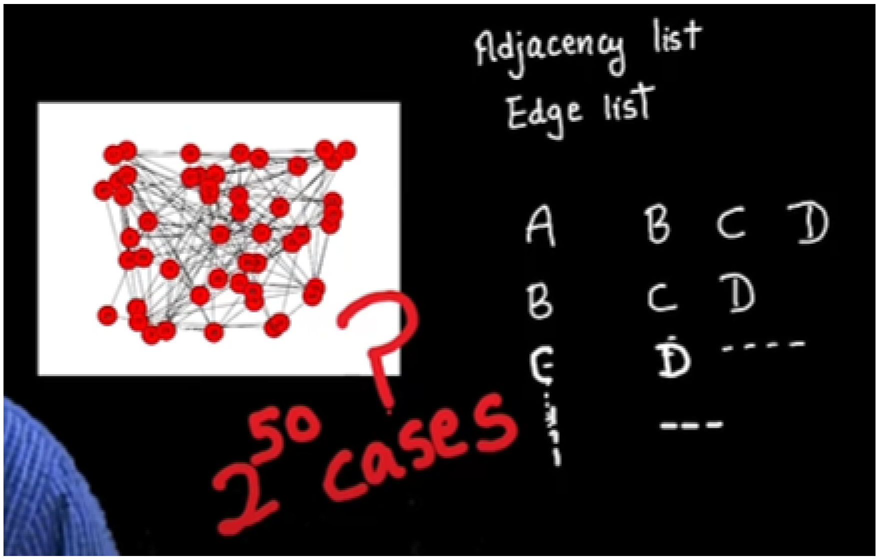
are a lot within them and a lot less outside this is technically called a bunch of nodes is

that and you call this partitioning as a valid community partitioning if such a partitioning is

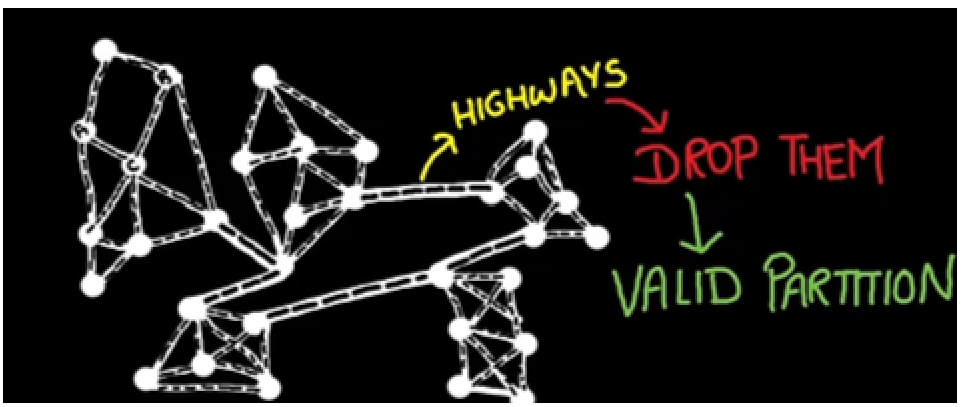
such that there is a lot of intercommunity edge sparse city and intra community edge density

intra means what within inter means between So, between it is less within it is more. Then

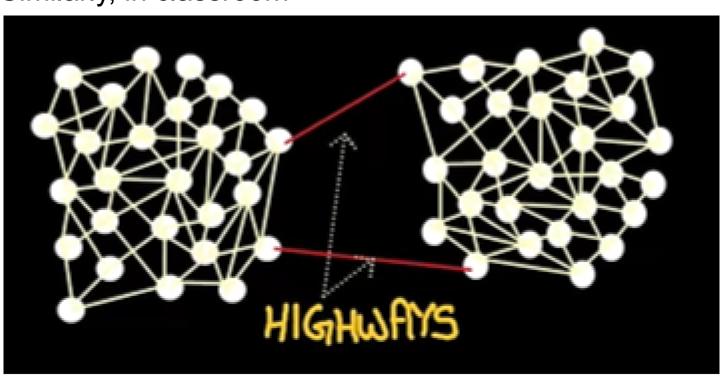
Very difficult to know by coding that how many grps exist



In a road ntwrk of cities and states, if u identify the highways(bridges) and drop them, then it leads to valid partition



Similarly, in classroom



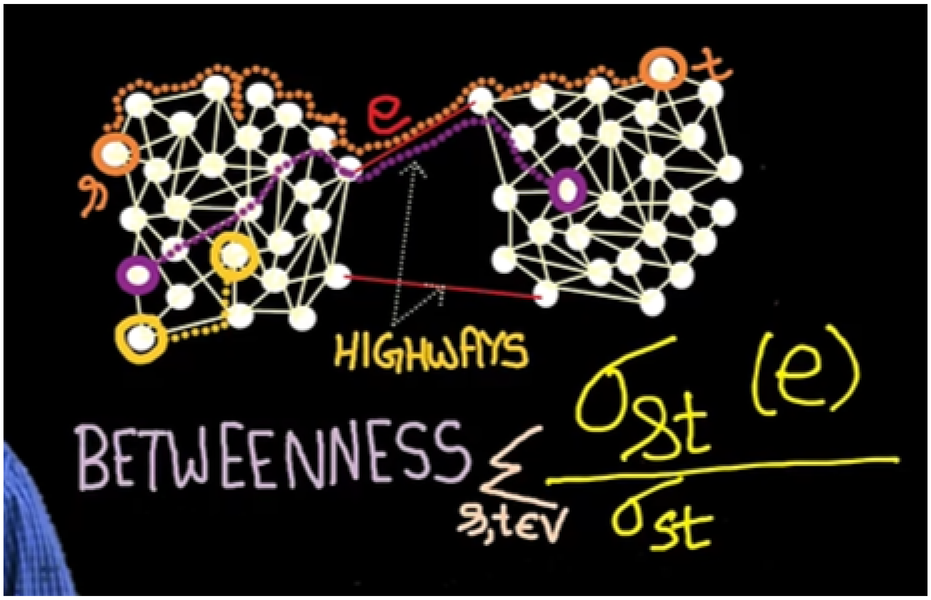
betweenness of an edge is defined as the total number of paths that go on it

Or

is the total number of paths that pass through

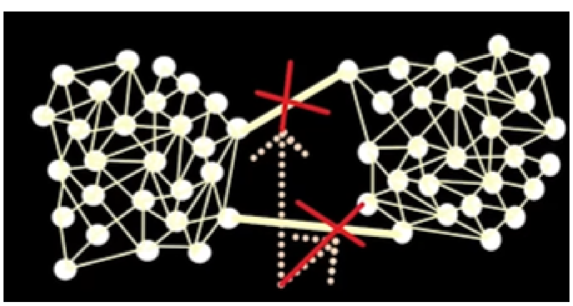
this edge the total number of shortest paths tha
pass through this edges versus the total number

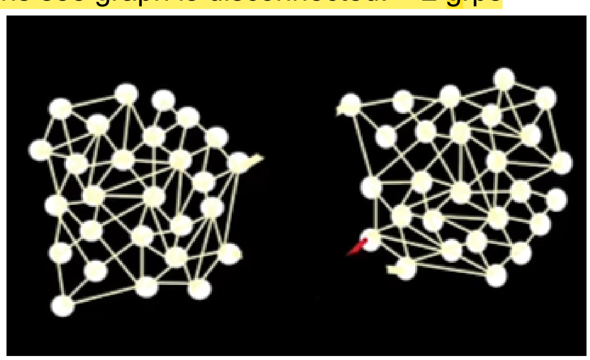
of shortest paths between two vertices what do I



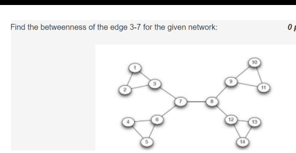
this higher the betweenness more does it connect
two components that otherwise are not connected

To check for communities - calculate betweeness of all the edges in classroom , and then u see 2 edges with high btwness, remove themm ans see graph is disconnected. = 2 grps



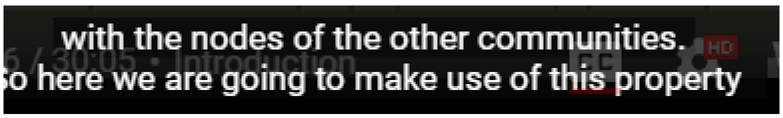


https://www.youtube.com/watch?v=IFky1eOeJs8



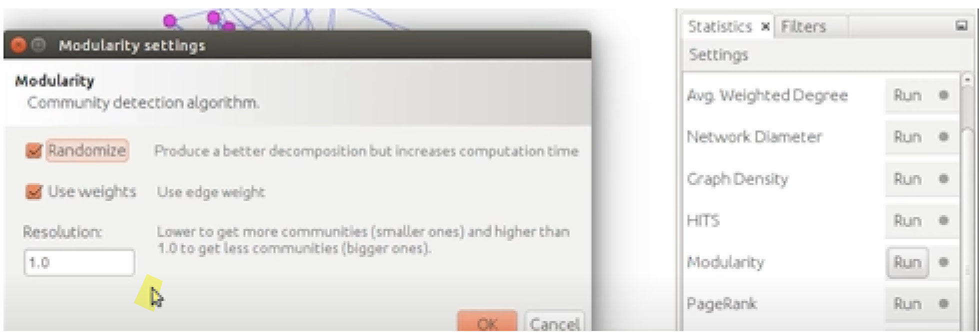
assignmnt

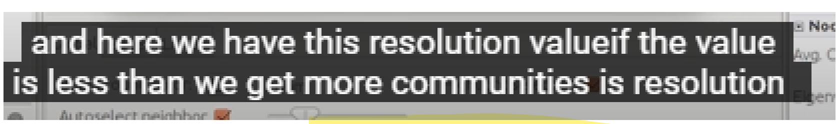
Lecture 37 - Finding Communities in a graph (Brute Force Method) - 1 == CODING VIDEO

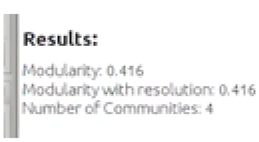


Lecture 38 - Community Detection Using **Girvan Newman Algorithm**
FINDING edge betweeness of all edges and removing the ones with max

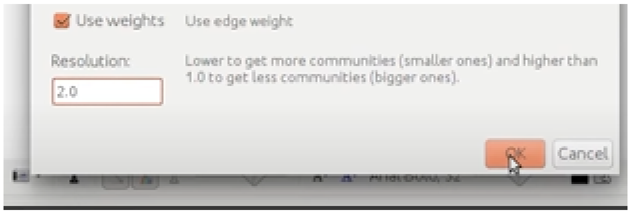
Lecture 39 - Visualising Communities using Gephi
We can see the communities by clicking on MODULARITY

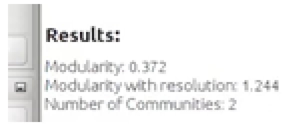




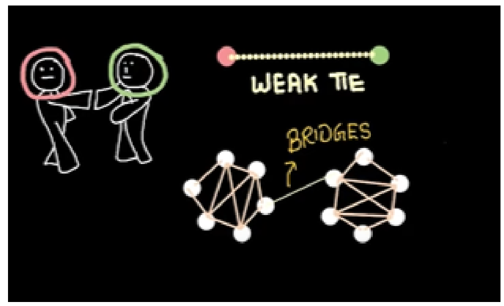


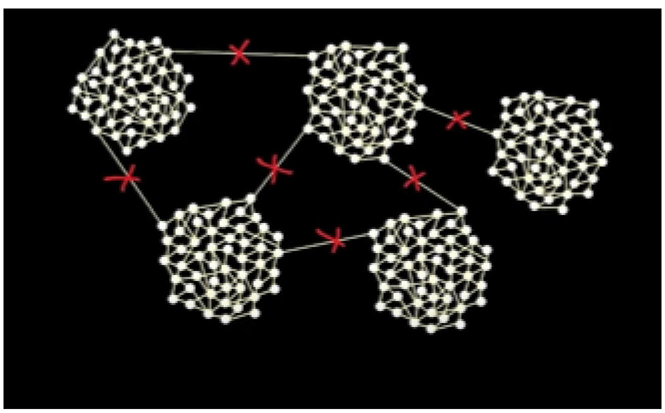
Changing resolution

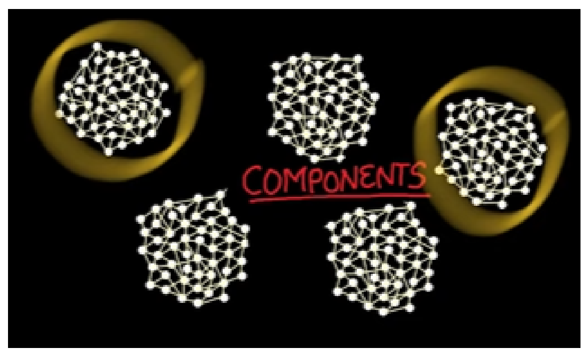


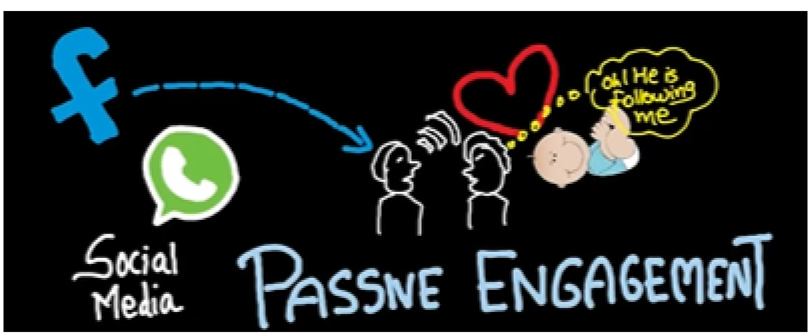


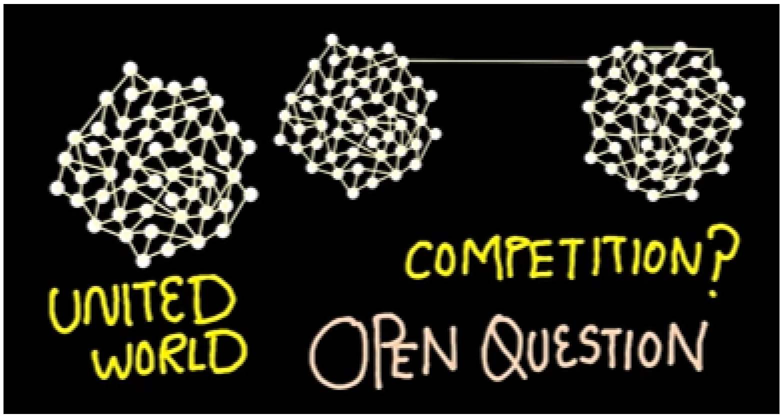
Lecture 40 - Strong and Weak Relationship - Summary
For jobs -







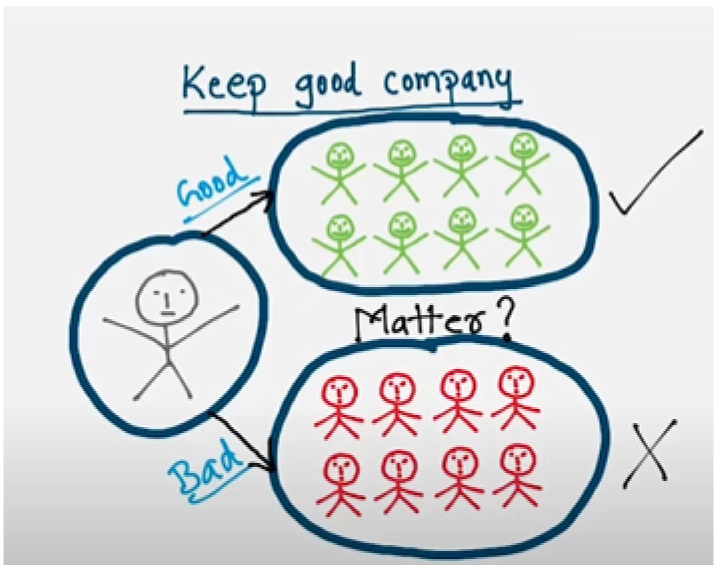


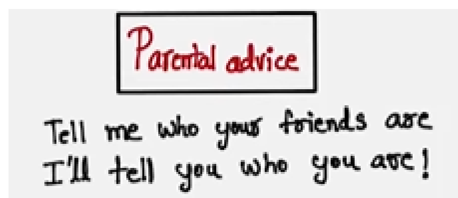


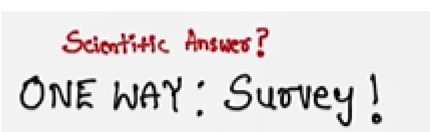
which is better ??

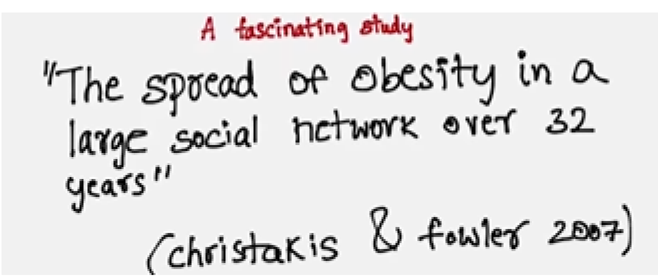
WEEK 4

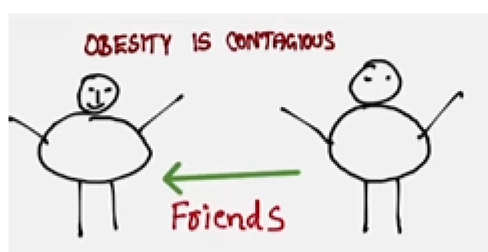
Lecture 41 - Introduction to Homophily - Should you watch your company ?

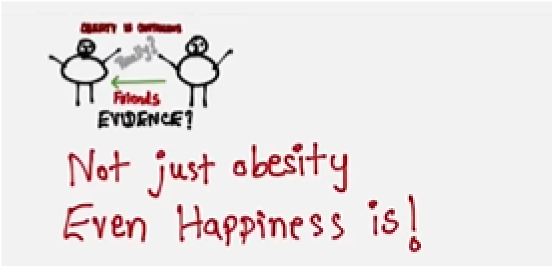


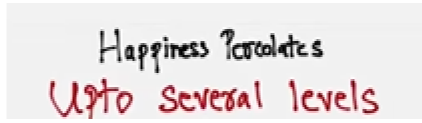




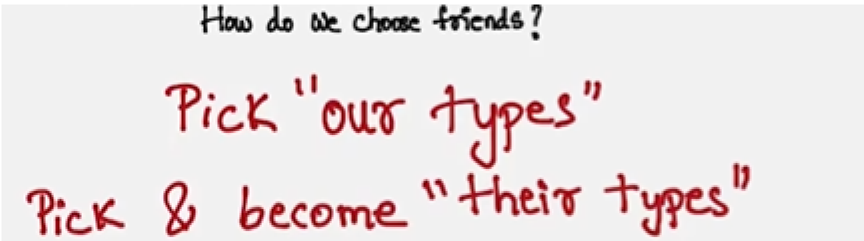




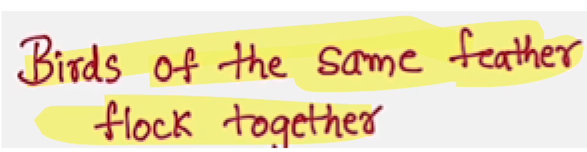




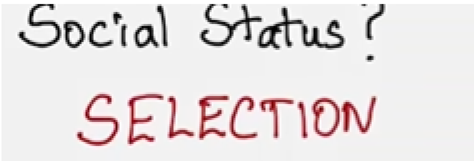
Lecture 42 - Selection and Social Influence

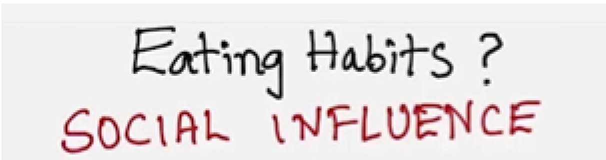


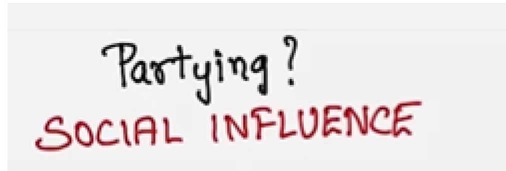
select friends thinking that they are our type



Smoking - social influence



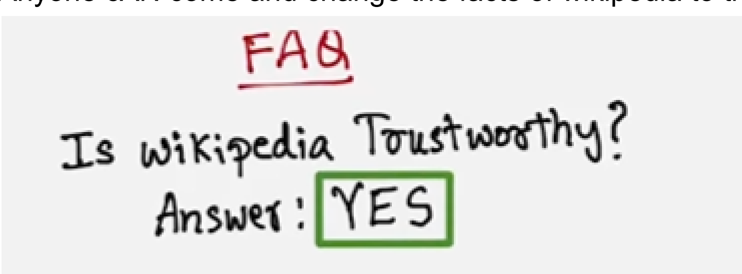




Lecture 43 - Interplay between Selection and Social Influence

Wikipedia dataset
two main ques -

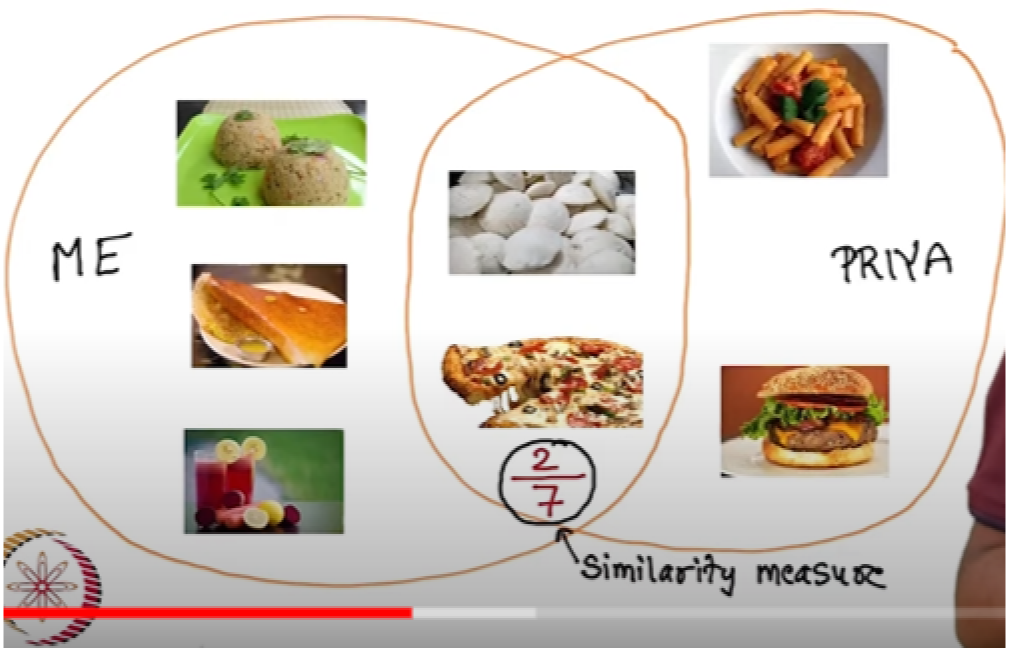
Anyone cAN come and change the facts of wikipedia to the right one



In the background - 

Dataset available → SELECTION or SOCIAL INFLUENCE ?

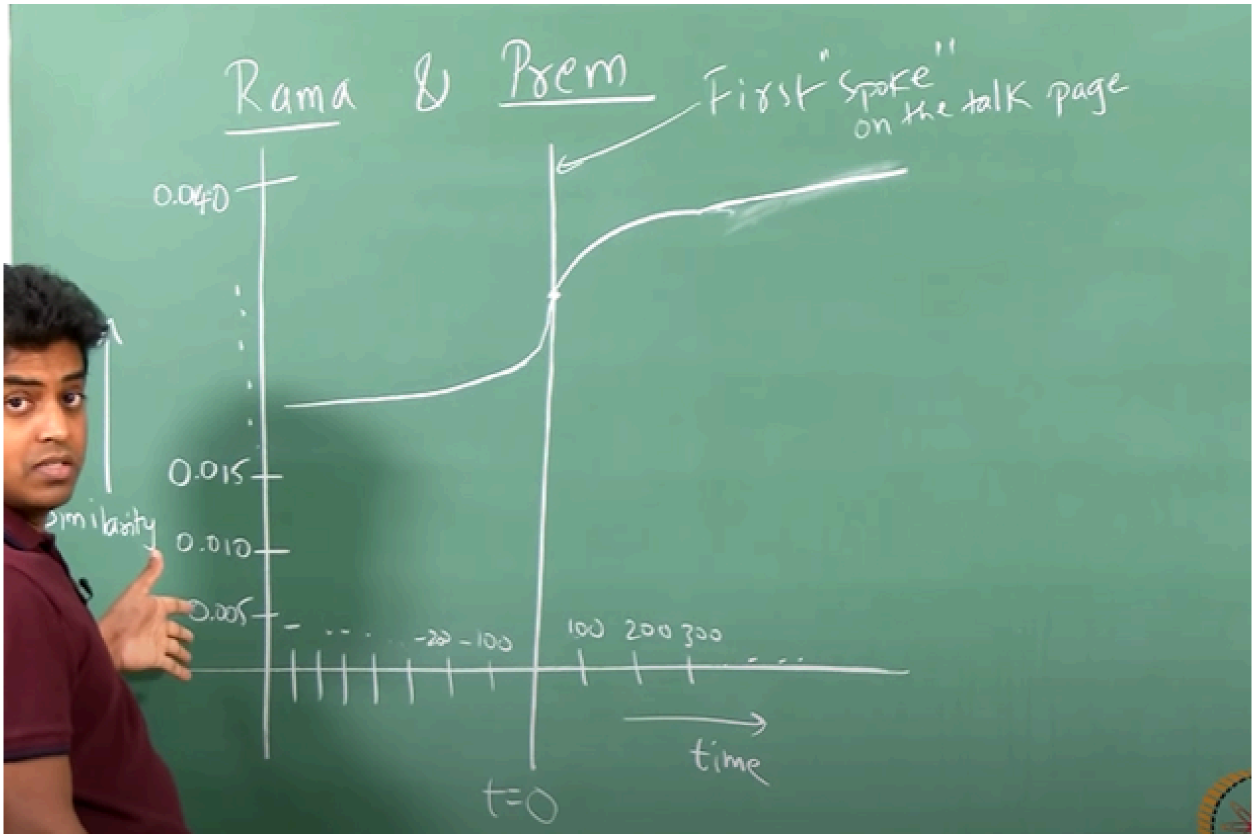
SIMILARITY MEASURE



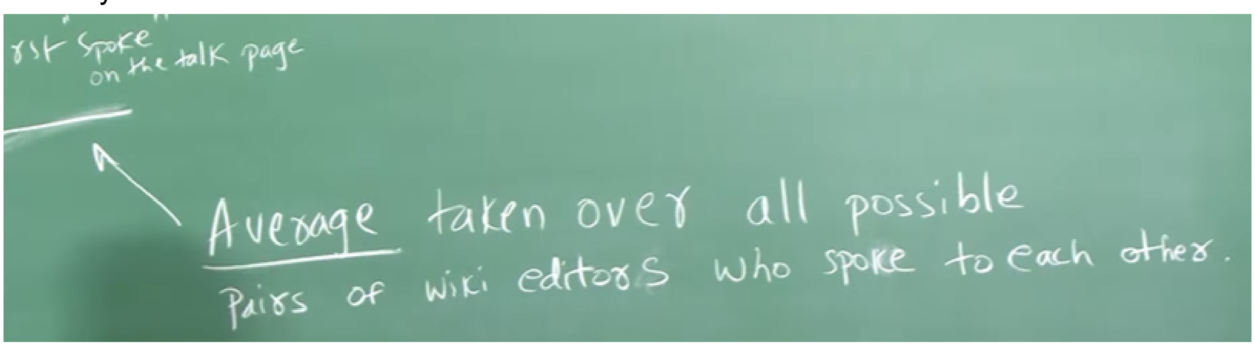
SIMILARITY MEASURE
WIKIPEDIA

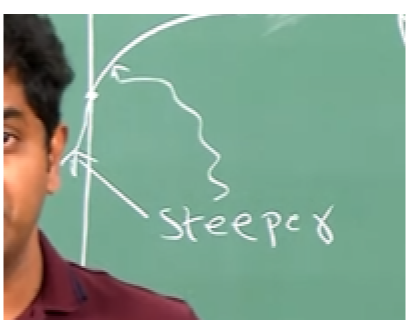
$$\frac{\text{Pages edited by both of us}}{\text{Total pages edited by both of us}}$$

2 people talking on wikipedia and they 1st time spoke at t = t0

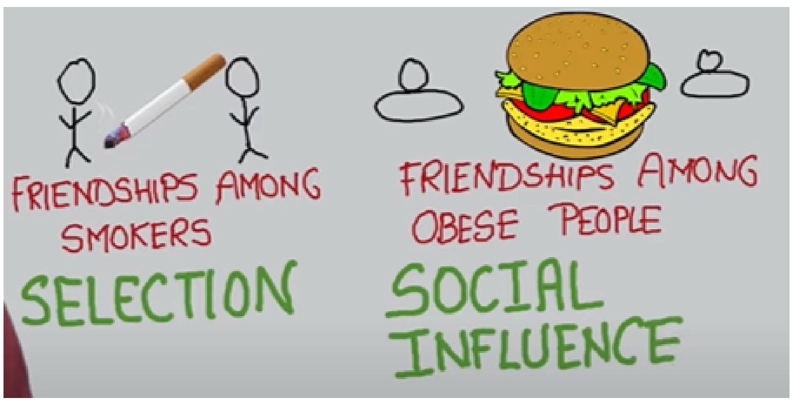


Since there similarity were common, they talked to each other and furthermore increase the similarity.





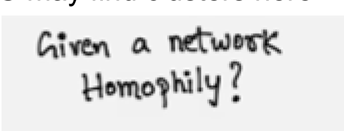
Acc to other some research -



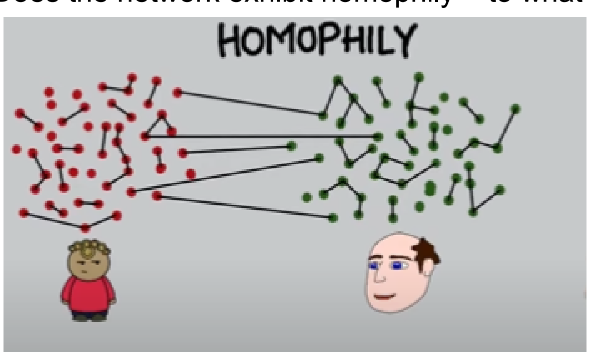
Lecture 44 - Homophily - Definition and measurement

A party - 50 middle age , 50 teenage

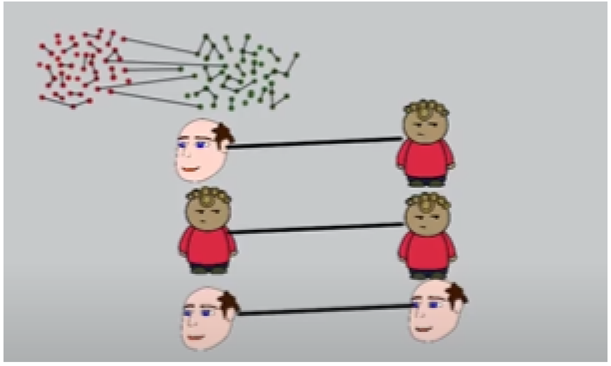
U may find clusters here

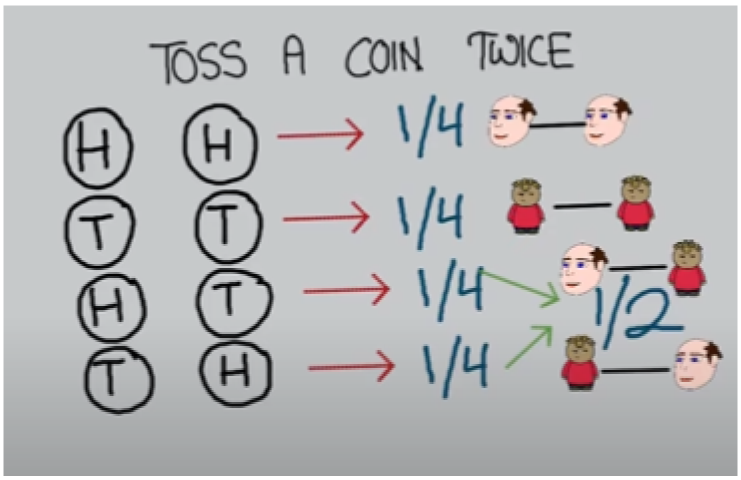


Does the network exhibit homophily + to what extent



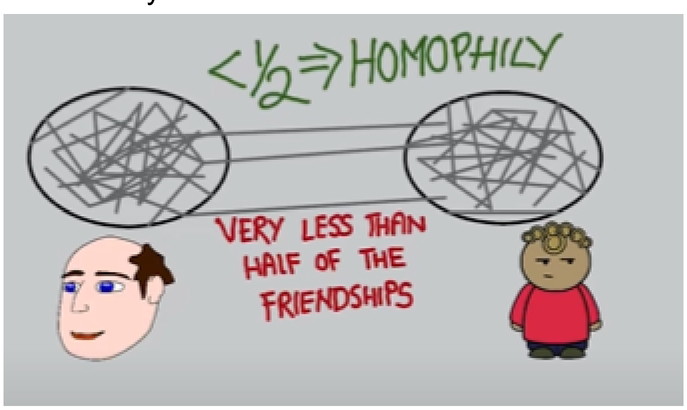
if u pick one edge from these, then what did u suspect its in btwn



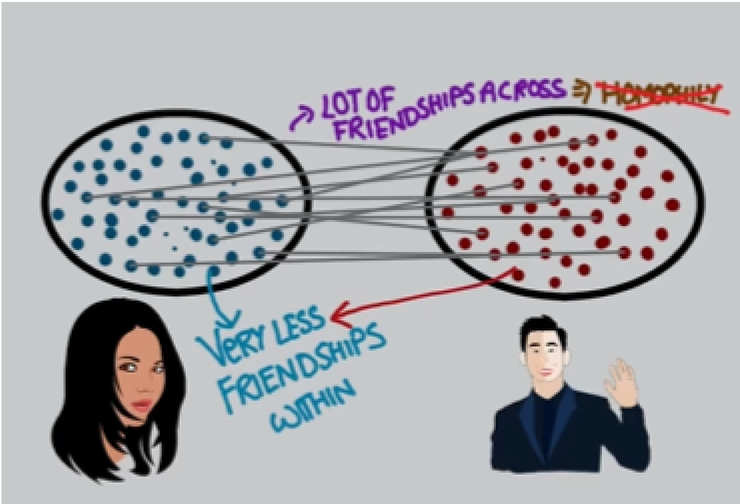


,

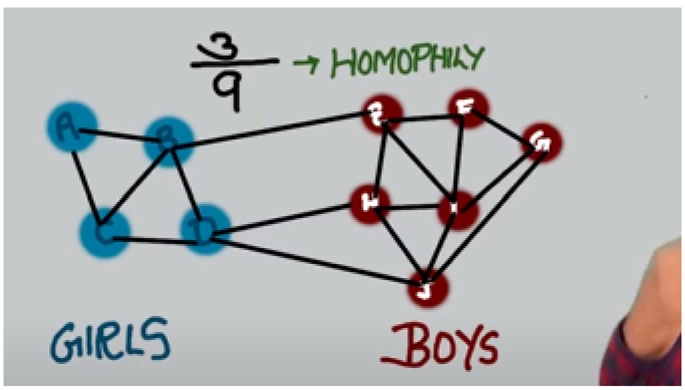
But in reality -

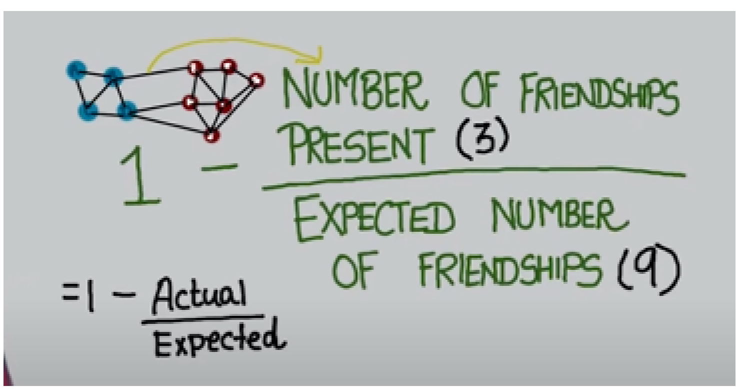


== party hall filled of actors and actresses

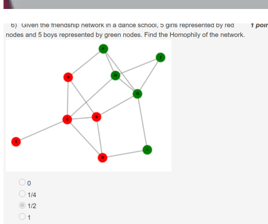


For checking homophily = actual friendship across / total friendships across (that should exist= total edges/2)
Lesser the fraction, more the homophily





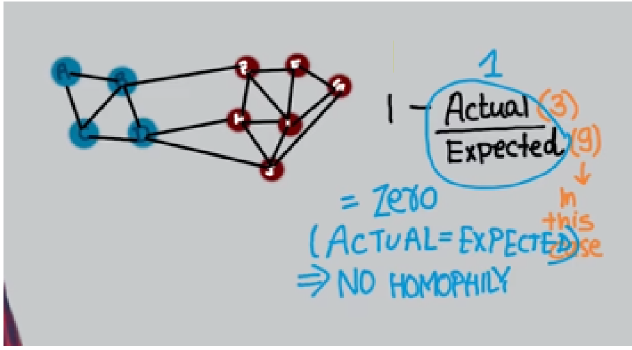
as the no. goes higher, higher homo

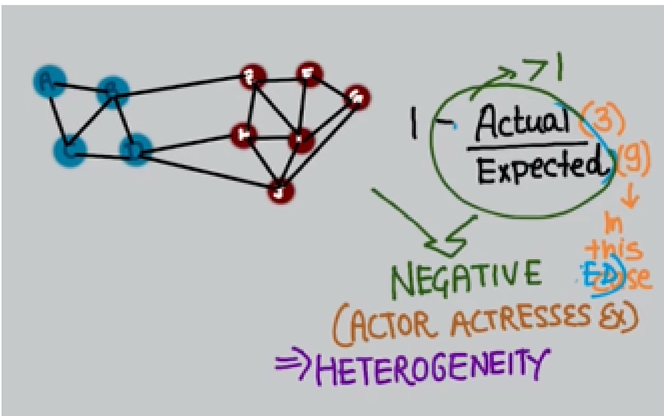


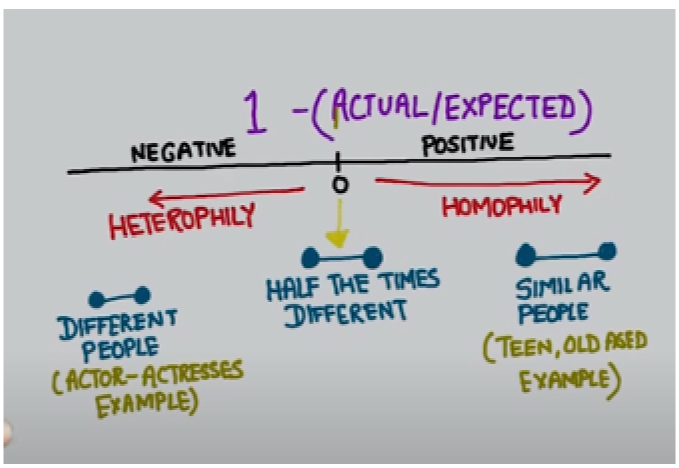
6) Given the friendship network in a dance school, 5 girls represented by red nodes and 5 boys represented by green nodes. Find the Homophily of the network. 1 poi

- 0
- 1/4
- 1/2
- 1

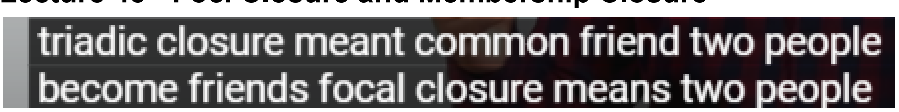
Yes, the answer is correct.

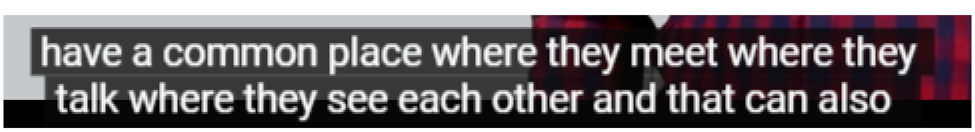


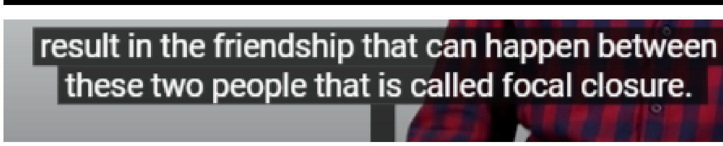


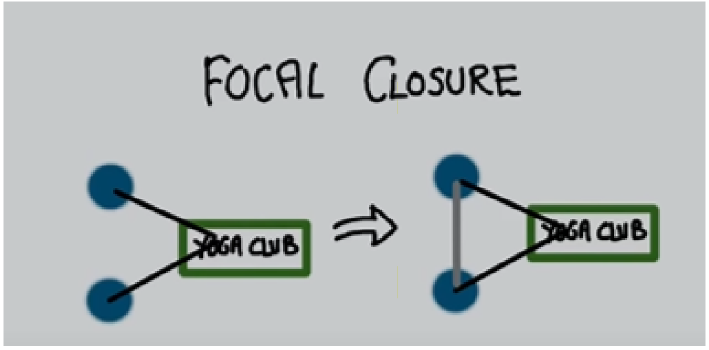


Lecture 45 - Foci Closure and Membership Closure







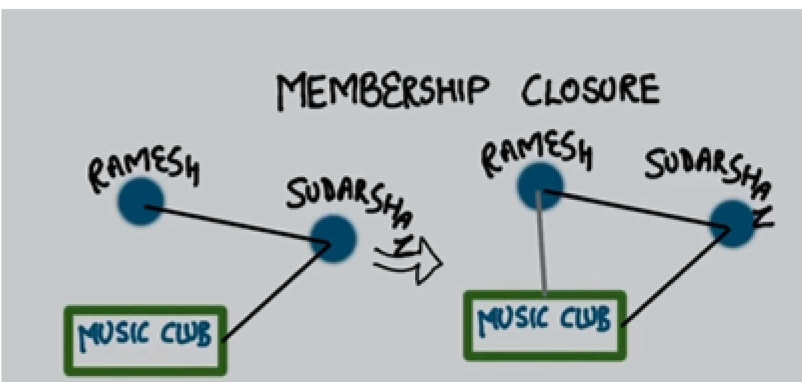


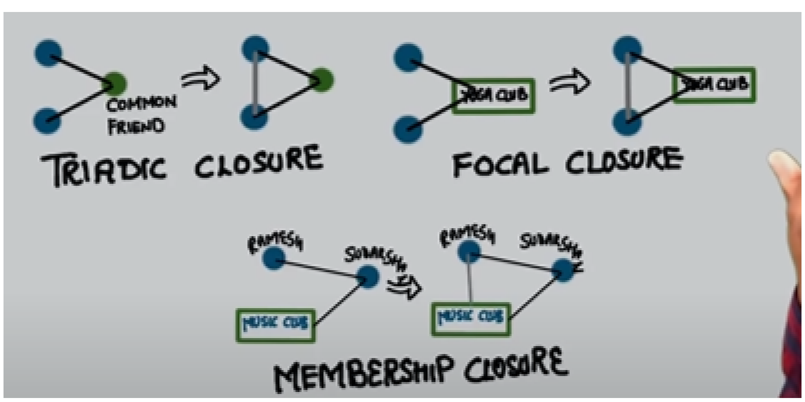
3 closures – TRIADIC , FOCAL . MEMBERSHIP CLOSURE

Ramesh are two people who know each other and I
go to the music club and now because he is friends

with me I will take him to the music club this
is called membership closure I make my friends

as members of this focal point which is let's
say music club so to per phrase we saw triadic

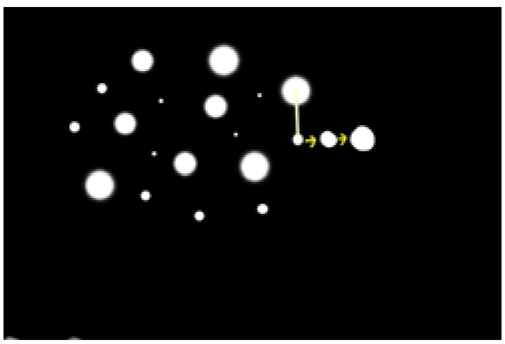


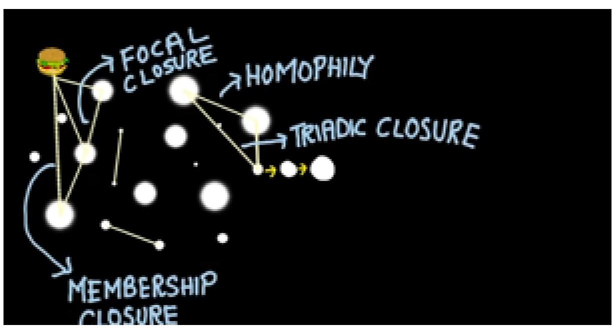


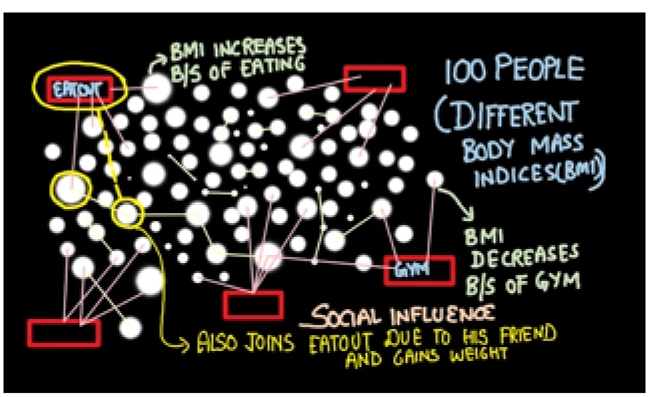
Lecture 46 -
Introduction to Fatman Evolutionary model

Main concepts -

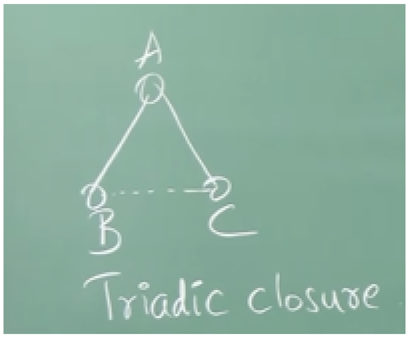
City - people with different body weights

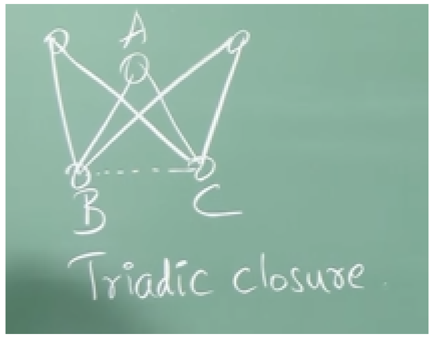




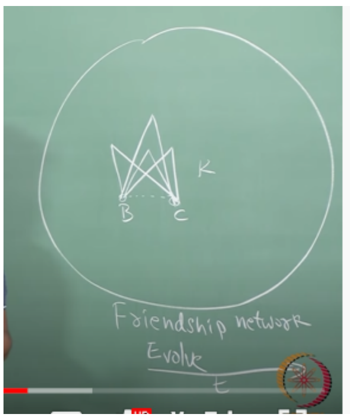


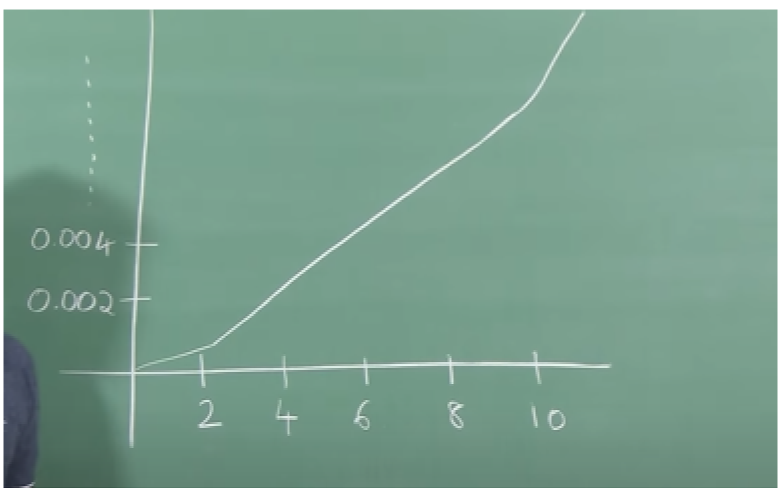
Lecture 50 - Quantifying the Effect of Triadic Closure

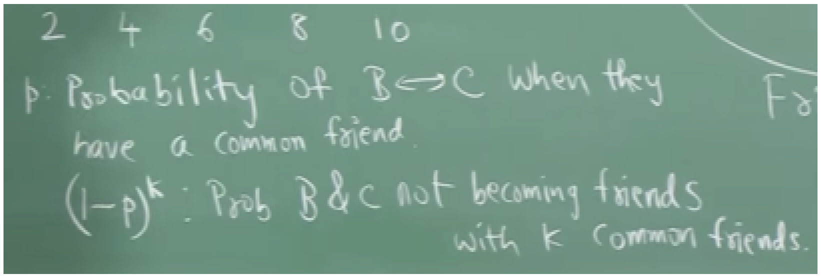


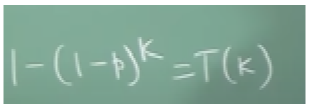


2 people having K(2,4,6…) no. of common friends , what is the prob of having a closure or becoming friends









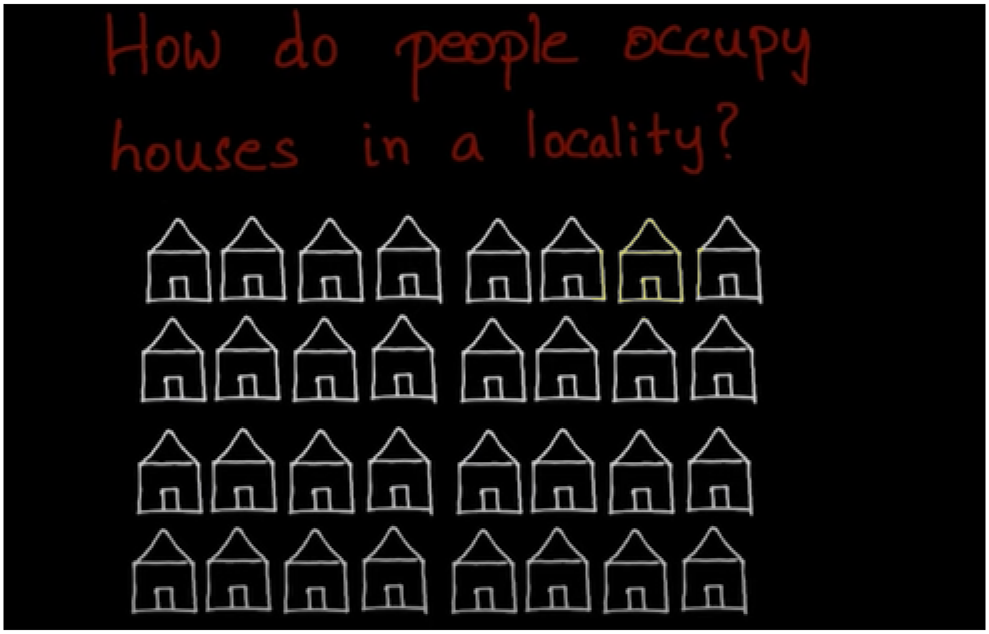
= prob of becoming frnd

- Lecture 45 - Foci Closure and Membership Closure
- Lecture 46 - Introduction to Fatman Evolutionary model
- Lecture 47 - Fatman Evolutionary Model- The Base Code (Adding people)
- Lecture 48 - Fatman Evolutionary Model- The Base Code (Adding Social Foci)
- Lecture 49 - Fatman Evolutionary Model- Implementing Homophily
- Lecture 50 - Quantifying the Effect of Triadic Closure
- Lecture 51 - Fatman Evolutionary Model- Implementing Closures
- Lecture 52 - Fatman Evolutionary Model- Implementing Social Influence
- Lecture 53 - Fatman Evolutionary Model- Storing and analyzing longitudnal data

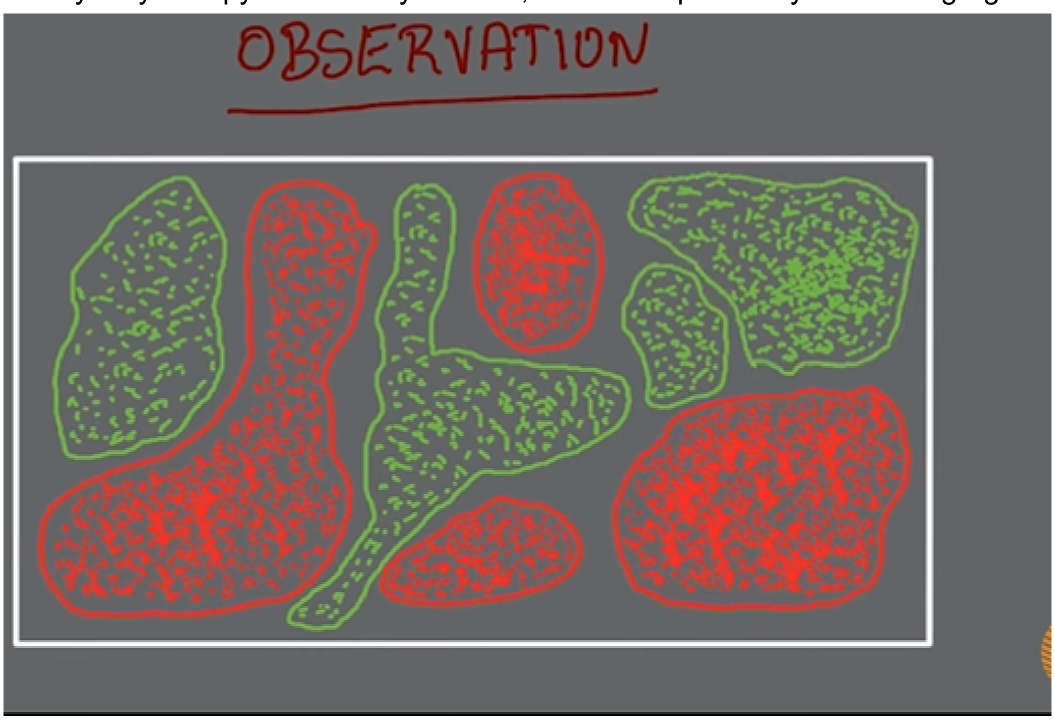
WEEK 5

Lecture-54: Spatial Segregation: An Introduction

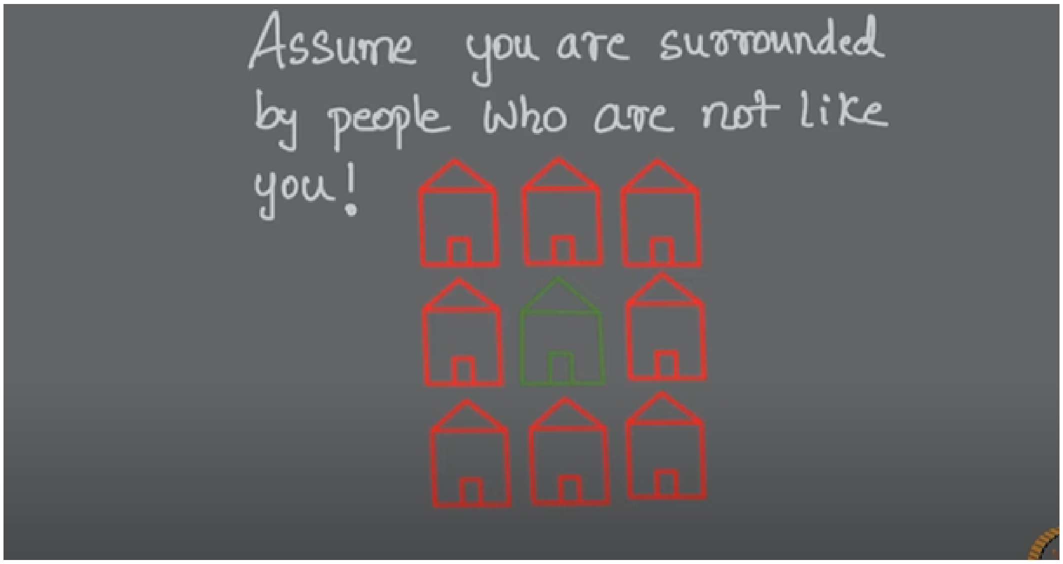
HOW do u choose your house to live in

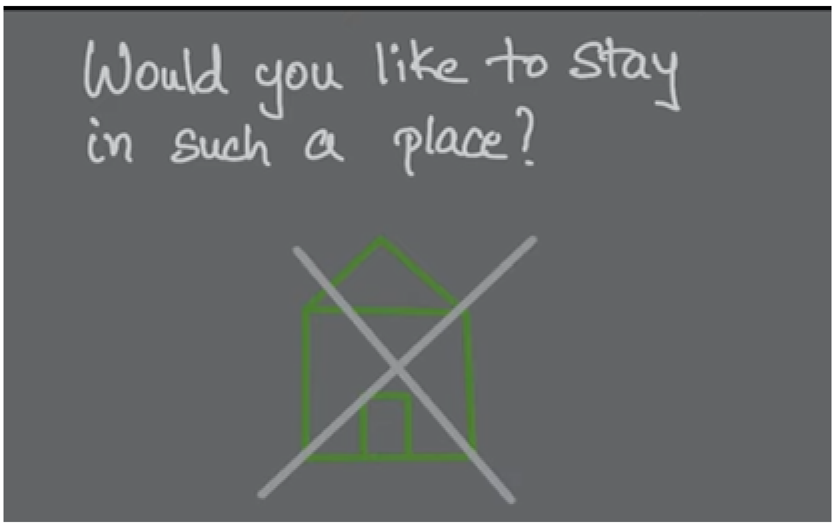


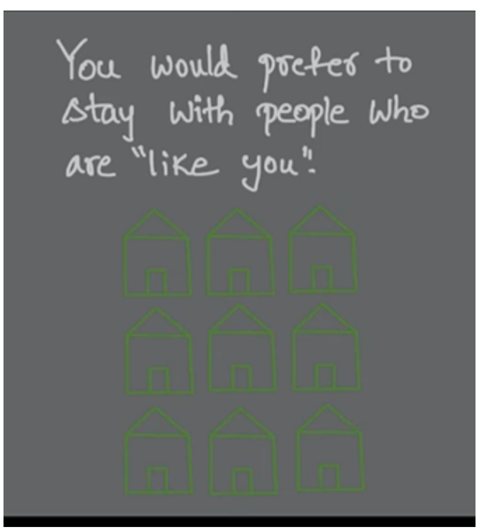
Initially they occupy at uniformly random, but after a point they start to segregate

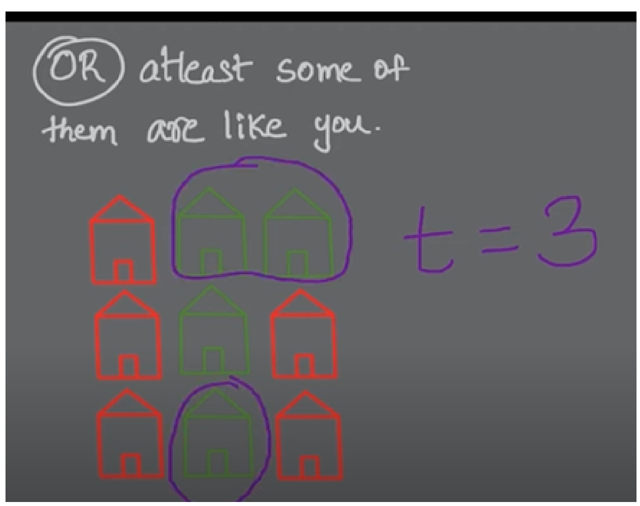


Like in foreign indians segregate

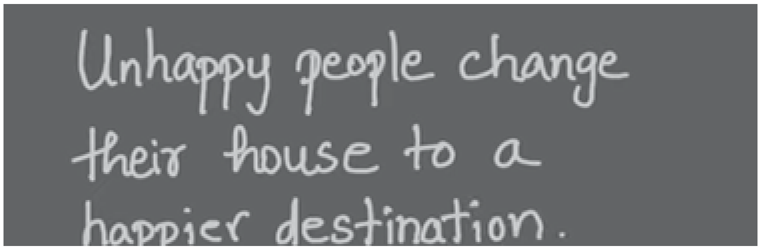






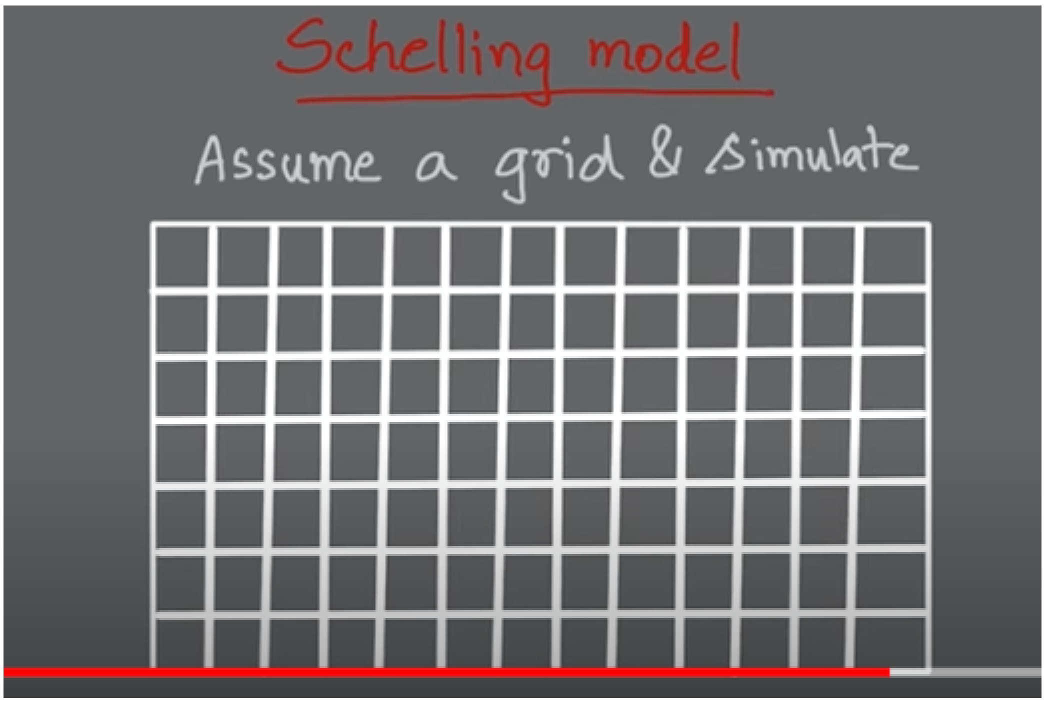


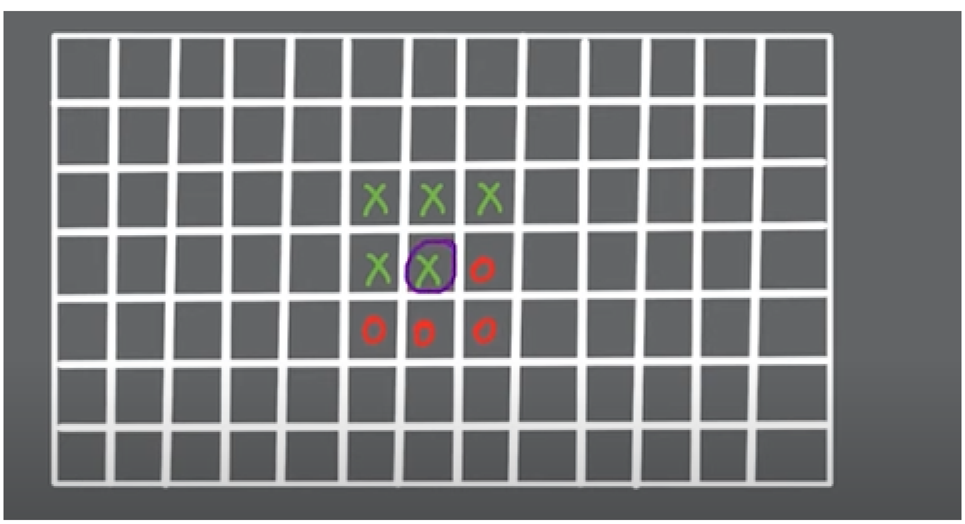
Whenerve you aare surrounfded by atleast 3 people like you =, then u choose ot stay there

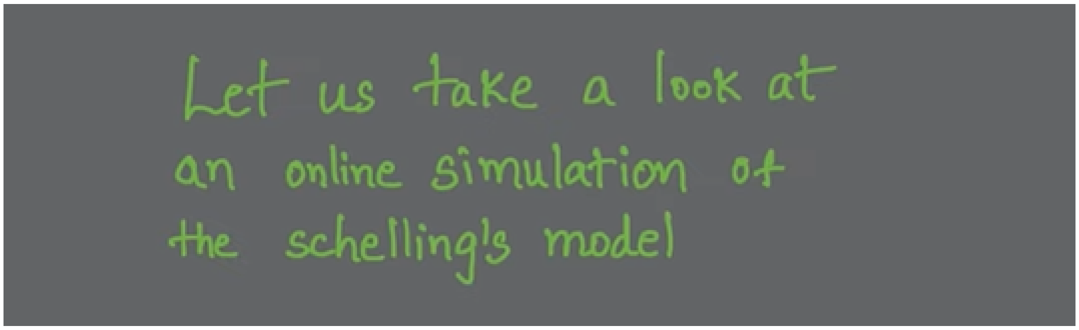


people with less than 3

CAN WE MODEL THIS ?? LEAVING THE DESTINATION. YES







schillings model simply says that

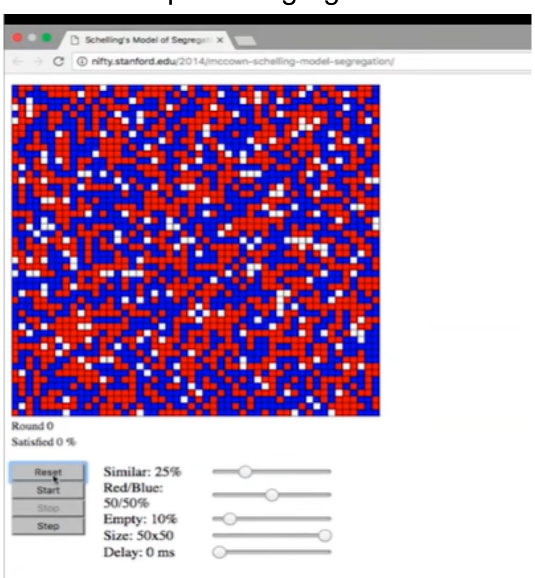
as and when people are unhappy because they are surrounded byneighbours below their threshold

value. What do I mean by that? Sounds a little jargonic. All I am trying to say is my threshold

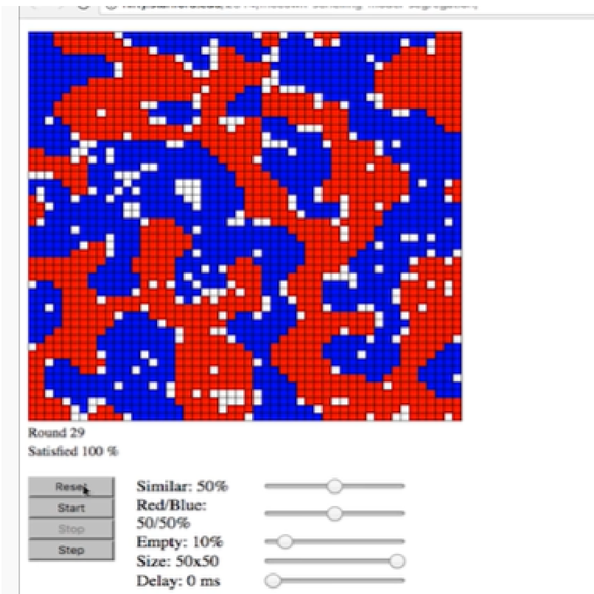
is three or may be two if I have two people in my neighbourhood who are of my type I will stay

there otherwise I will move let me take a big grid and then try simulating this and seeing

Lecture55- Spatial Segregation: Simulation of the Schelling Model

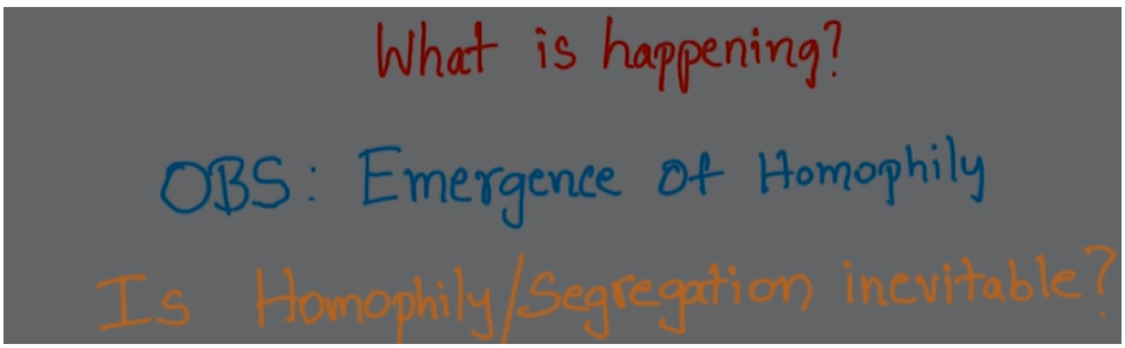


SIMILAR = 25% => if someone has 8 neighbours , he should have (25%) 2 neigjbours of his type to stay

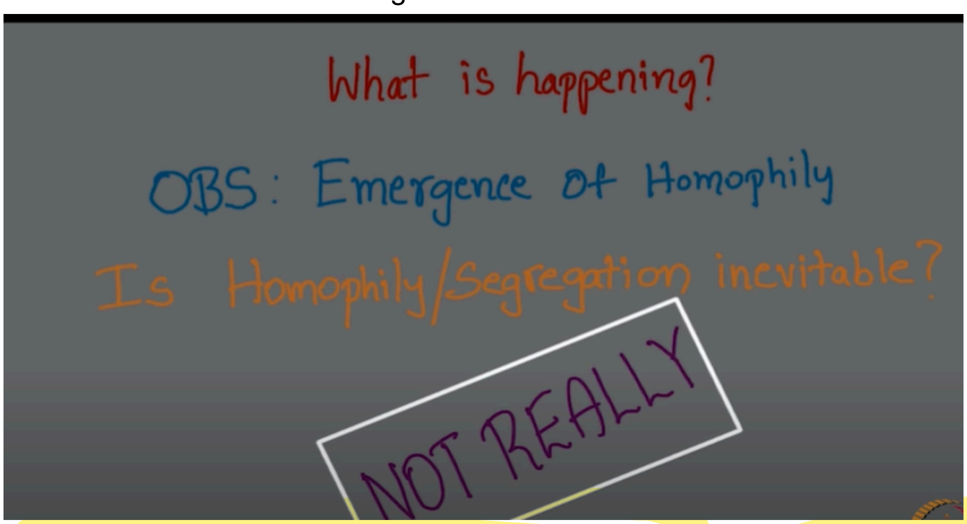


here 50prcnt is threshold

Lecture56- Spatial Segregation: Conclusion'



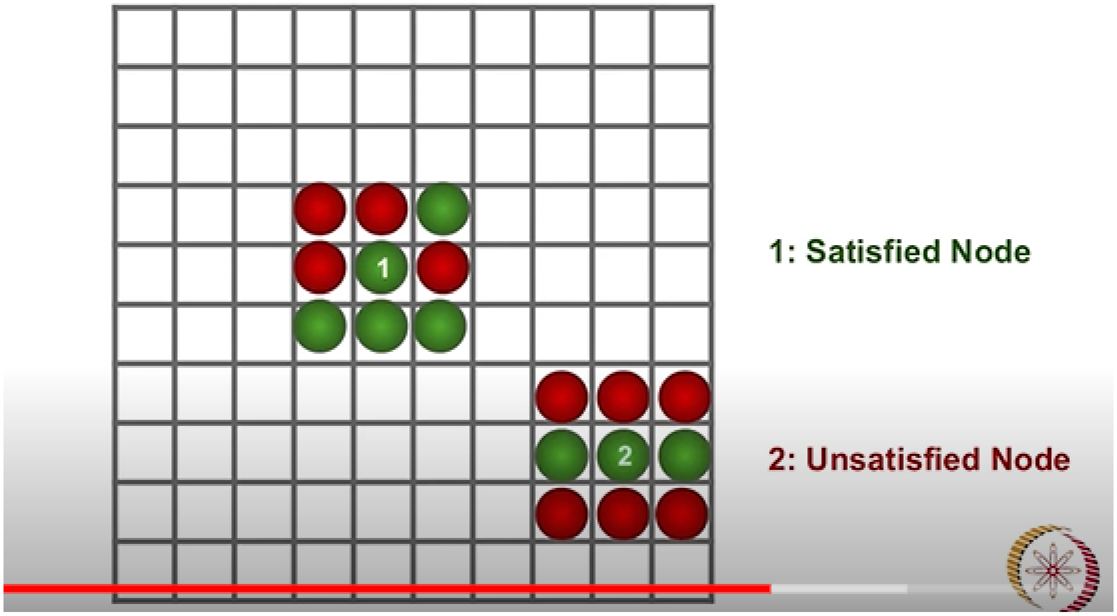
A local behaviour results in a global behavior



Spatial segregation doesn't always induce communities/clustering in the grid

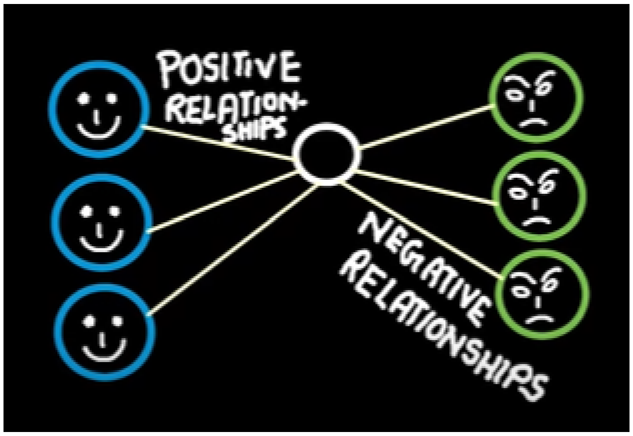
Lecture 57- Schelling Model Implementation-1(Introduction)

if T=3 ->



Unsatisfied move to new locations and grid changes and this is done for many iterations

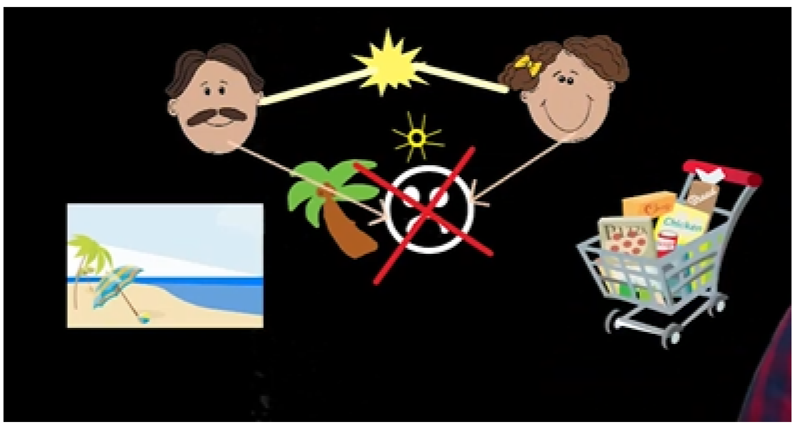
Lecture 62- Positive and Negative Relationships - Introduction

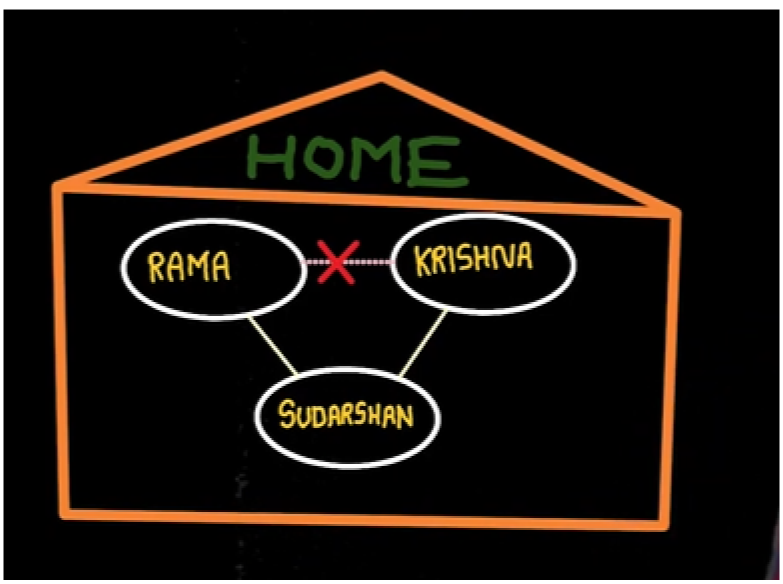


This is philosophically - now understanding the mathematics of this

Lecture 63- Structural Balance

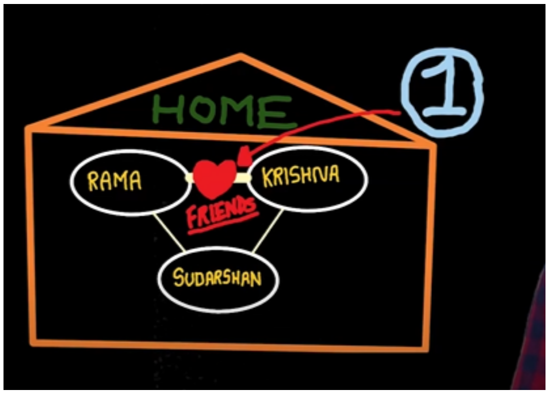
Uncle aunt fighting, give different options, i prefer to go with none safest option



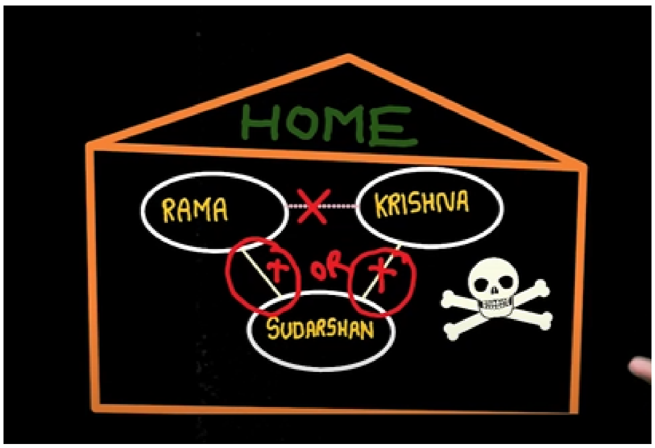


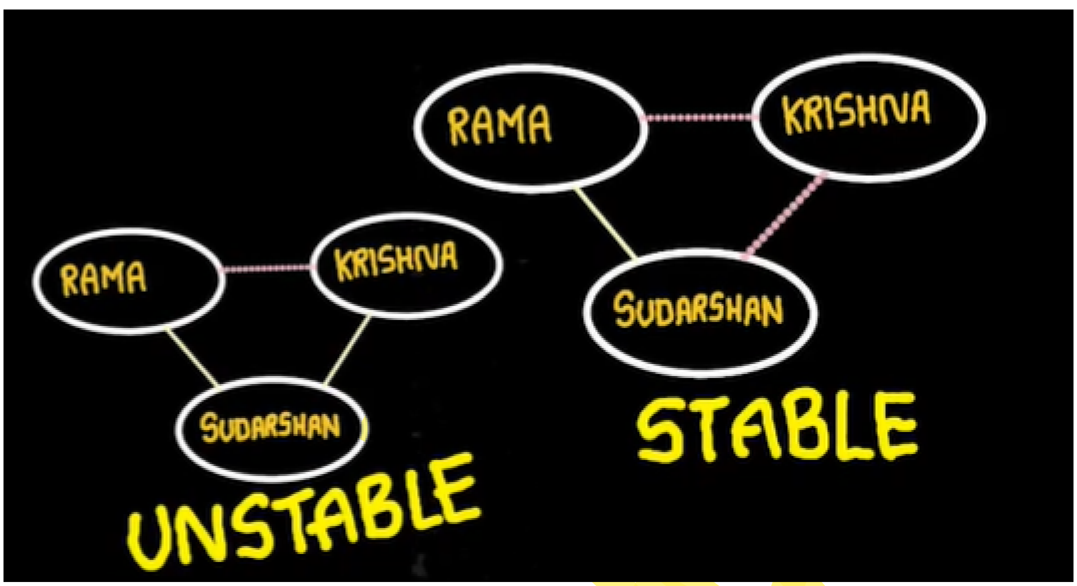
2 friends not liking each other

2 options - make them friends

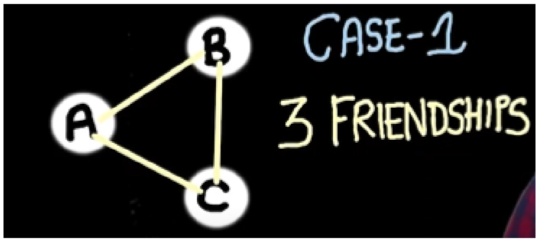


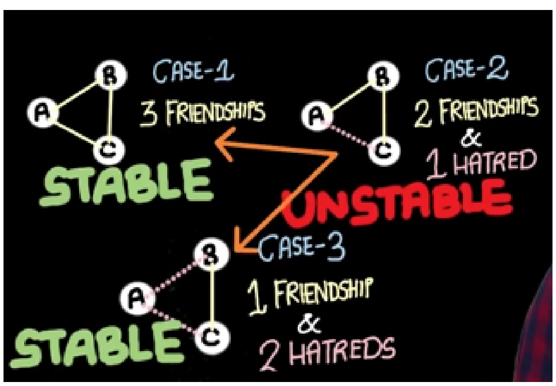
Or other case, all of u would not be friend if triadic closure couldnt happen

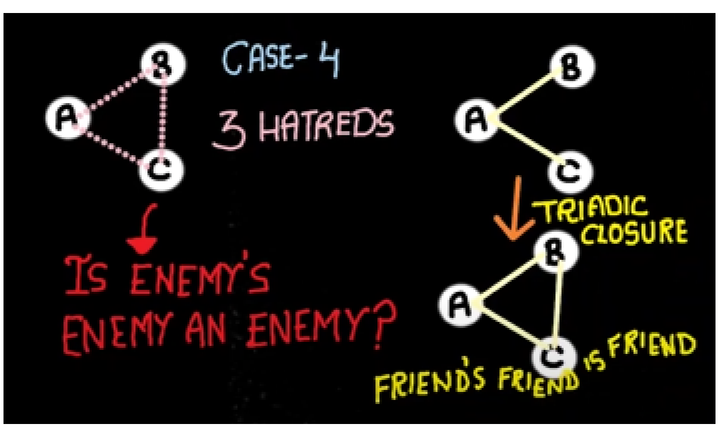




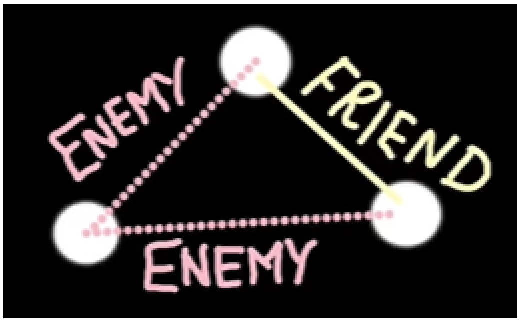
ALL CASES BETWEEN 3 FRNDS





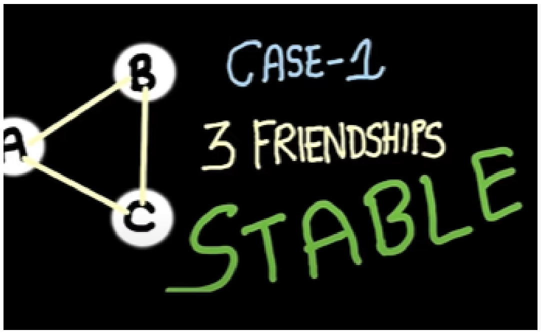


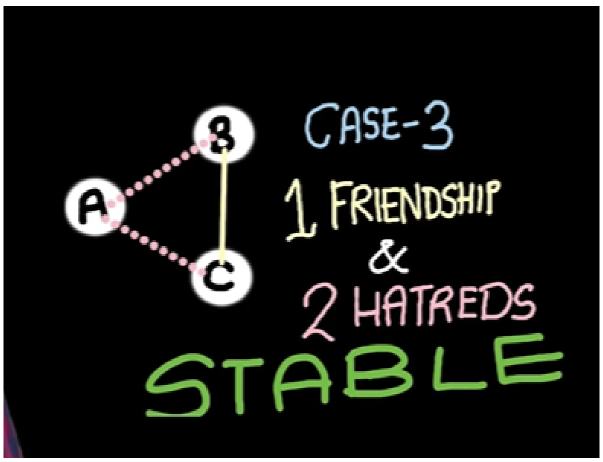
Lecture 64- Enemy's Enemy is a Friend



WHN U struggle / sweat together , ur bonding increases

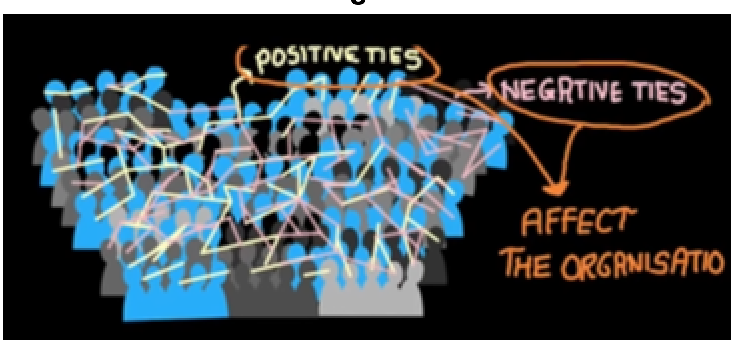
2 get a same enemy and become friends

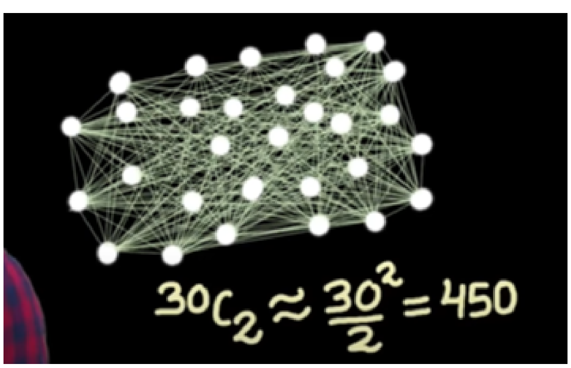




.

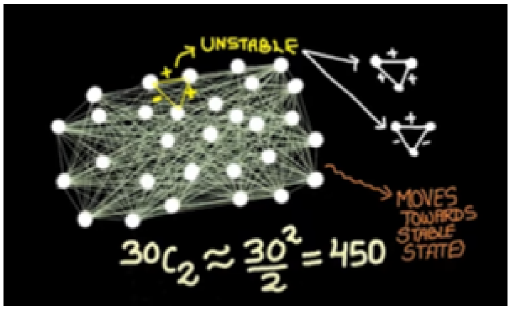
Lecture 65- Characterizing the structure of balanced networks

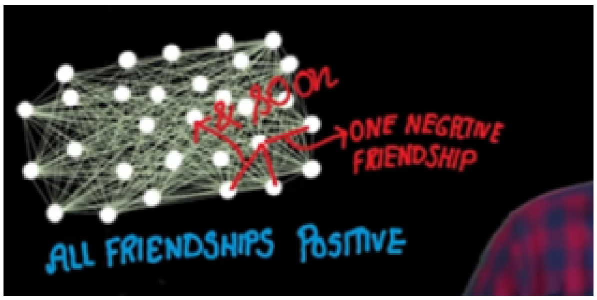




in office 30 people, all know each other.

Unstable triangle moves to stable state



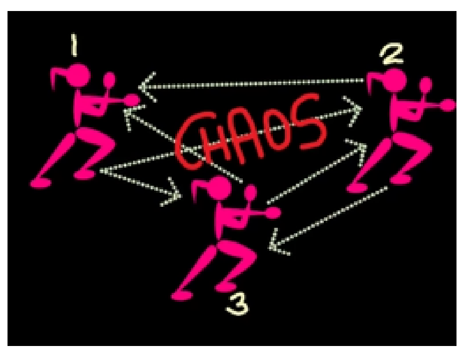


the negative friendship result in more negative frndsips

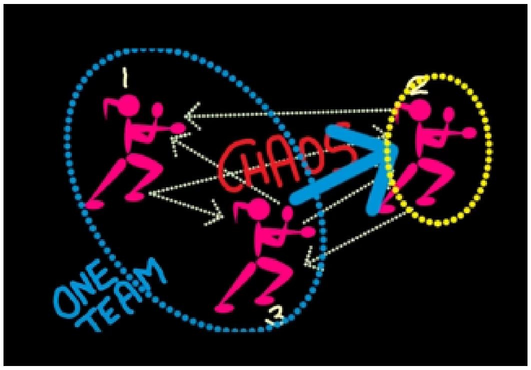
next lecture

Lecture 66- Balance Theorem

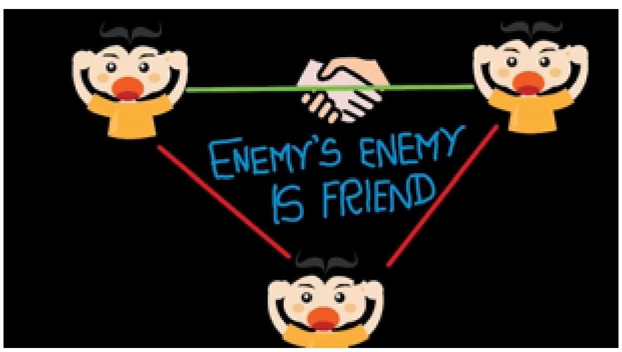
If 3 people in boxing ring, if one person fights the other 2, its a chaos

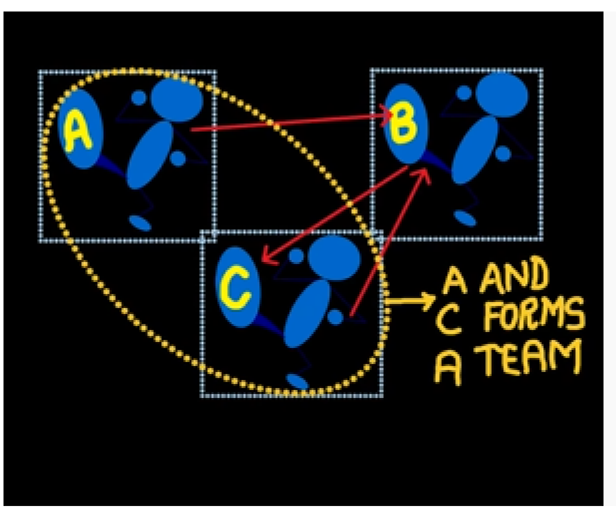


2 people need to team up -

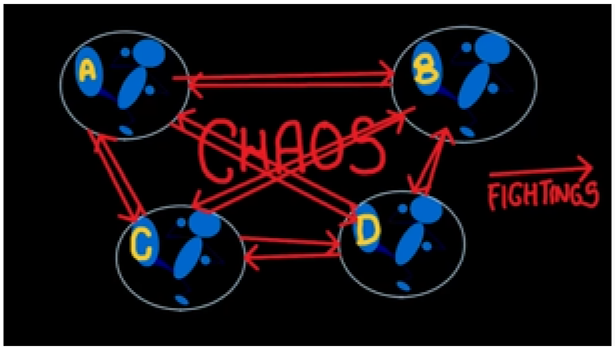


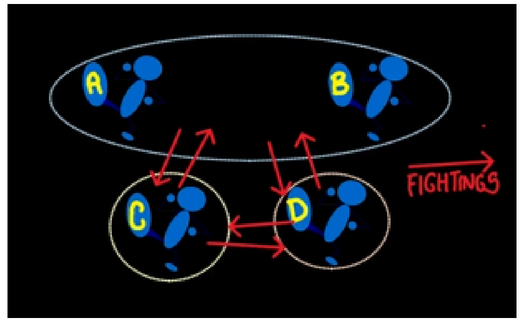
Reason ?? –

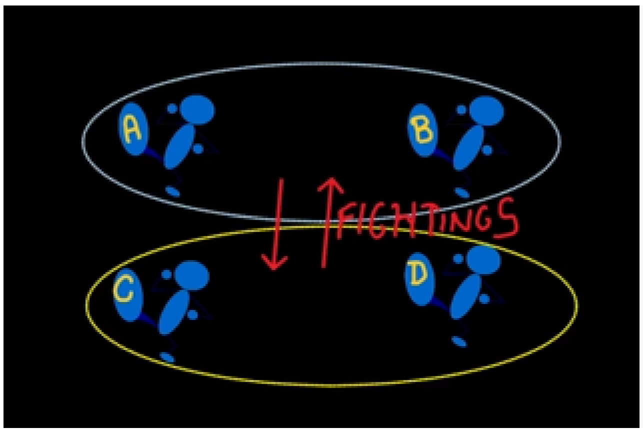




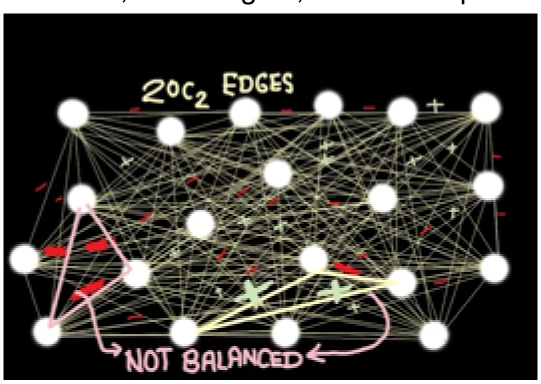
In case of 4 people

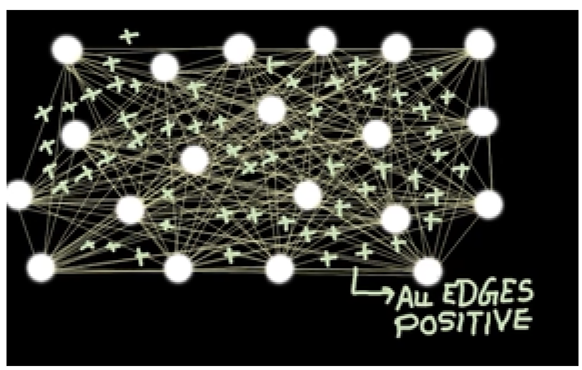


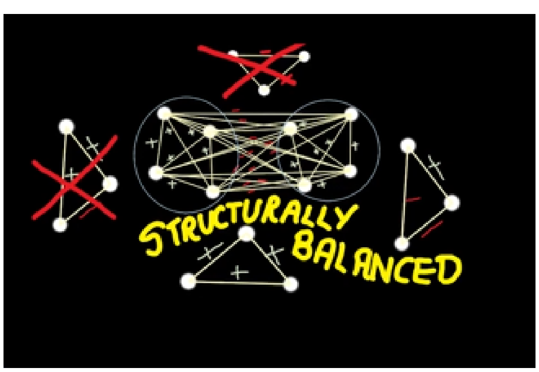




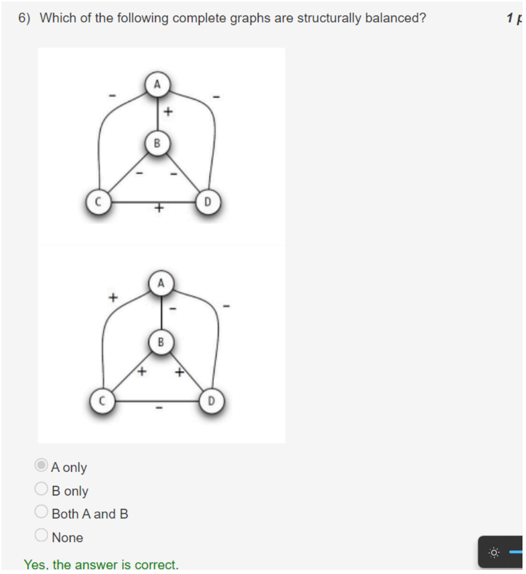
Lecture 67- Proof of Balance Theorem
20 nodes, 20c2 edges , + and - rships



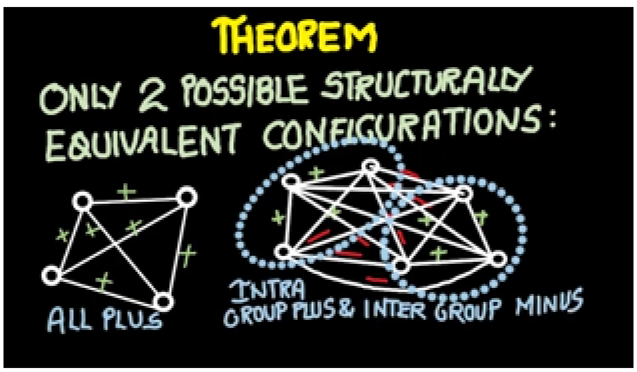




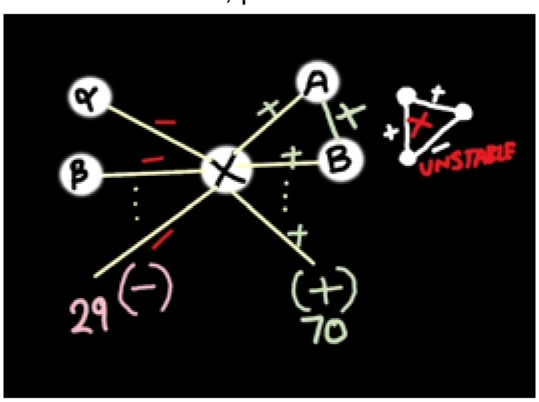
2 clusters forming a team with + withing and - across

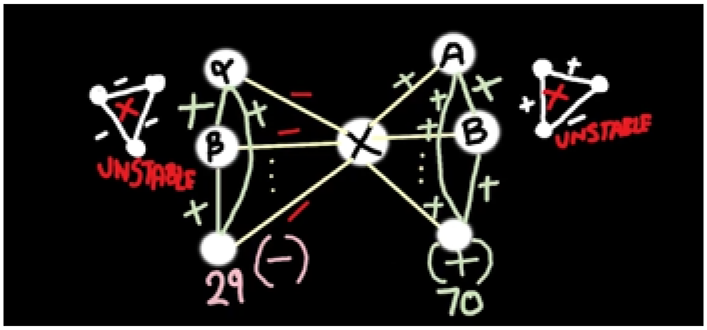


STRUCTURAL BALANCE MEANS aapas me sb positive ho or across me negative

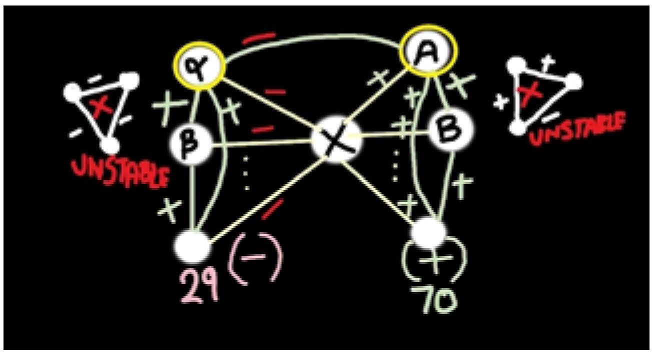


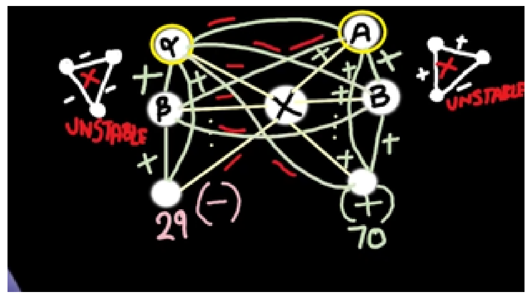
Out of 100 nodes, pick a node X -

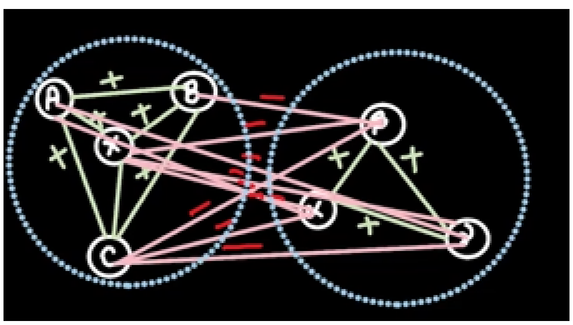




we observe thay among + of X all are +ve
Adn among   all - of X all are positive to maintain the stability
Now between a ,alpha and x, A and alpha has -ve relation





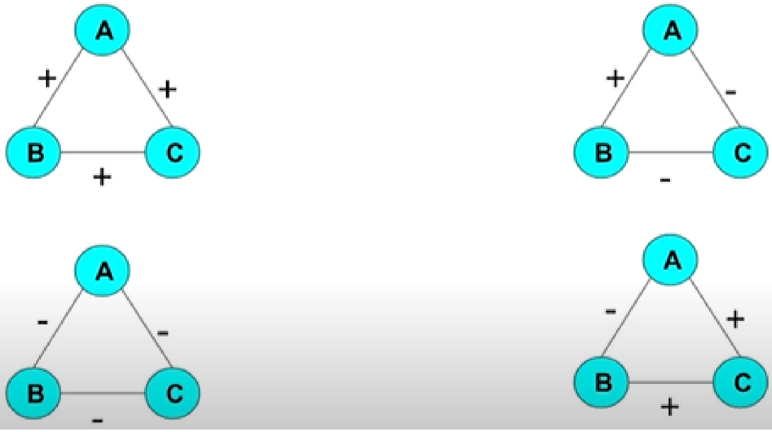


clusters are formed with all + inside and all - across

Lecture 68- Introduction to positive and negative edges

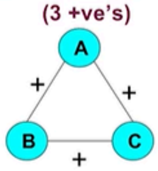
Signed Networks

Possible Triangles



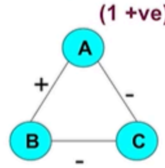
Stable Triangles

Three Mutual Friends
(3 +ve's)



(1)

Two friends with a common enemy
(1 +ve)



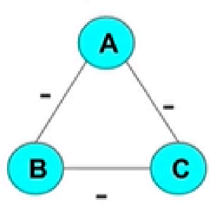
(2)

"The friend of my friend is my friend"

"The enemy of my enemy is my friend" (A for B)
"The enemy of my friend is my enemy" (B for C)
"The friend of my enemy is my enemy" (C for A)

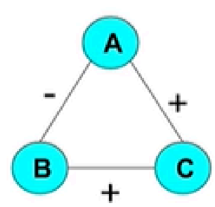
Unstable Triangles

Three Mutual Enemies
(0 +ve)

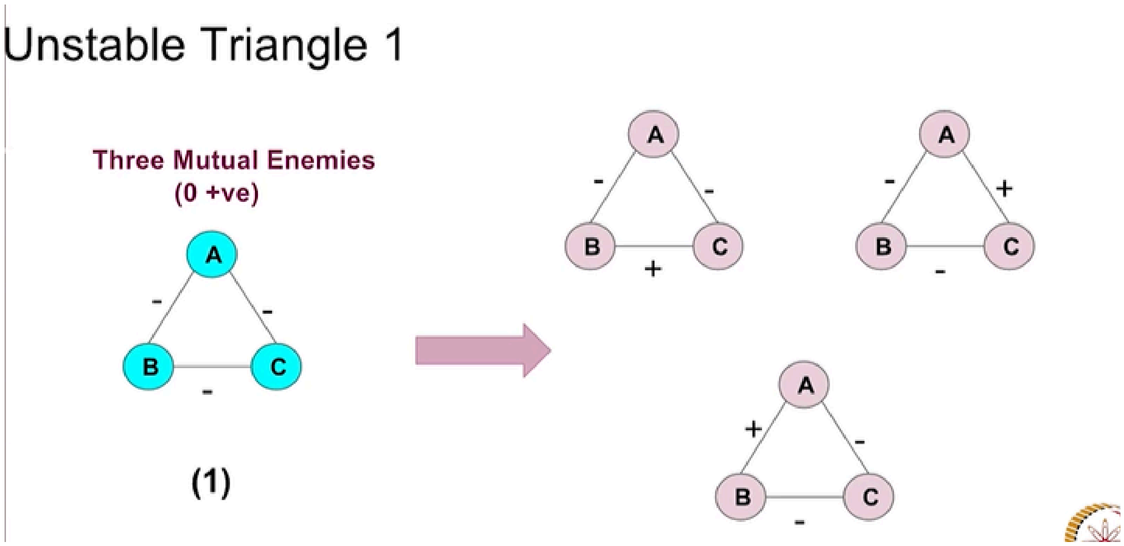


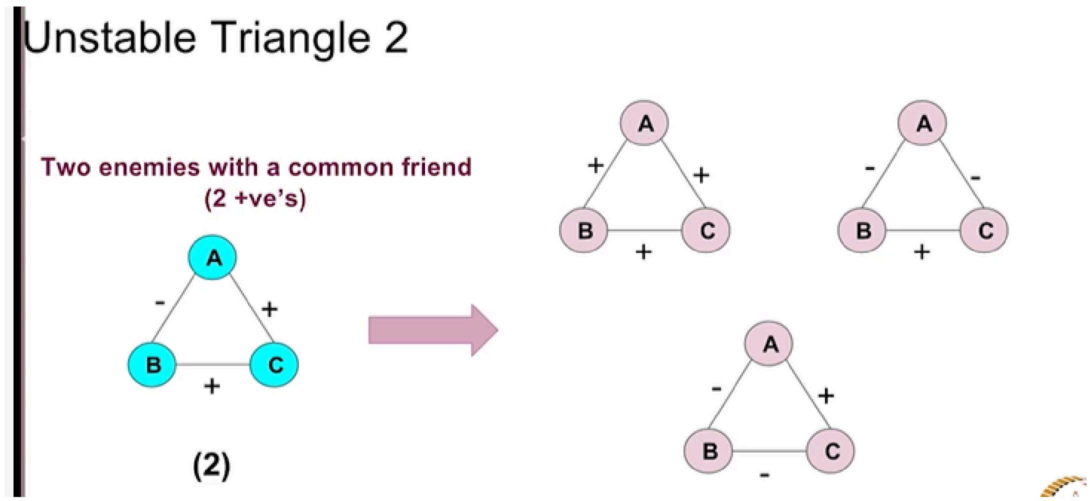
(3)

Two enemies with a common friend
(2 +ve's)



(4)





Lecture 69- Outline of Implemantation

Steps for Implementation

1. Create a graph with 'n' nodes, where the nodes are the countries.

2. Make it a complete graph by adding all possible edges. Also, assign '+' or '-' signs as weights to all the edges randomly.

3. Display the network.

4.1 Get a list of all the triangles in the network.

4.2 Store the sign details of all the triangles.

4.3 Count the number of unstable triangles in the network.

5. While the number of unstable triangles is not zero, do the following:

5.1. Choose a triangle in the graph that is unstable.

5.2. Make that triangle stable.

5.3. Count the number of unstable triangles

6. Now that there is no unstable triangle in the network, it can be divided into two coalitions, such that in each coalition, the intra-edges are positive, and the inter-edges are negative.

For doing point 6 : 6.1. Choose a random node. Add it to the first coalition.

6.2. Also put all the 'friends' of this node in the first coalition.

6.3. Put all the 'enemies' of this node in the second coalition.

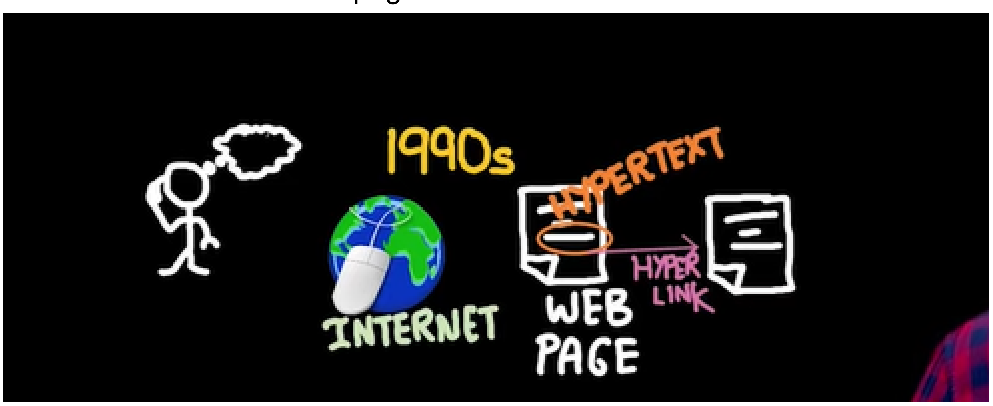
6.4. Repeat steps 6.2 and 6.3 for all the 'unprocessed' nodes of first coalition.

7. Display the network with coalitions

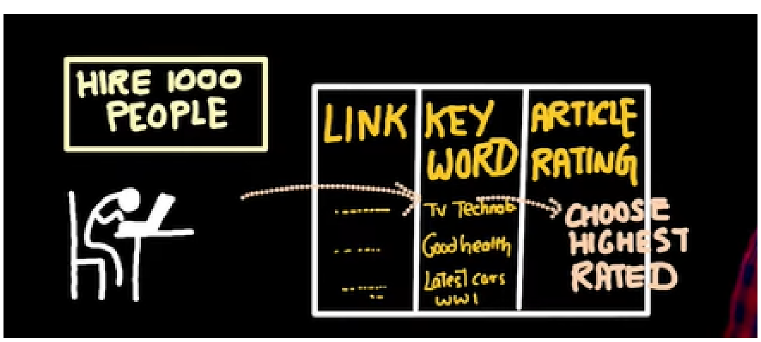
CODING VIDEOS –

- Lecture 70- Creating graph, displaying it and counting unstable triangles
- Lecture 71- Moving a network from an unstable to stable state
- Lecture 72- Forming two coalitions
- Lecture 73- Forming two coalitions (Continued)
- **Lecture 74- Visualizing coalitions and the evolution**

WEEK 6 - lecture 75 THE WEB GRAPH
Imagine we are 1990s, internet - developing, web page - containing hypertext, hyperlink - a link that takes u to another webpage



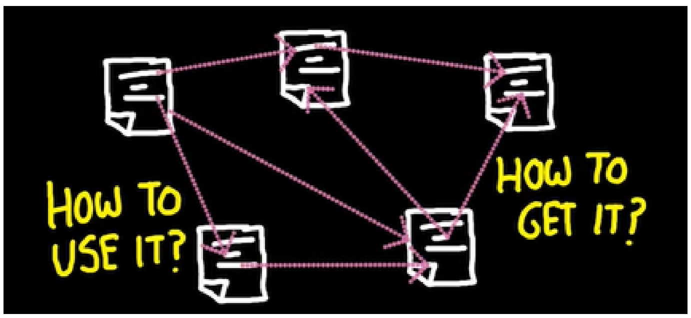
HOW TO LOOK FOR 1000S OF WEB PAGES HOW TO KNOW WHICH PAGE HAS BEST INFO ?



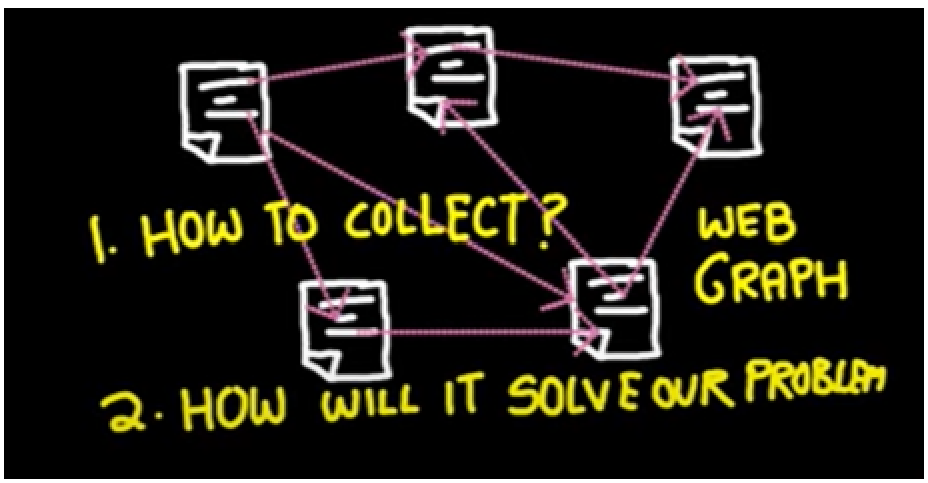
BAD IDEA

GAVE IDEA OF WEB GRAPH

WEB GRAPH - make each web page a node,and link denotes a page linking to other pages via hyperlink



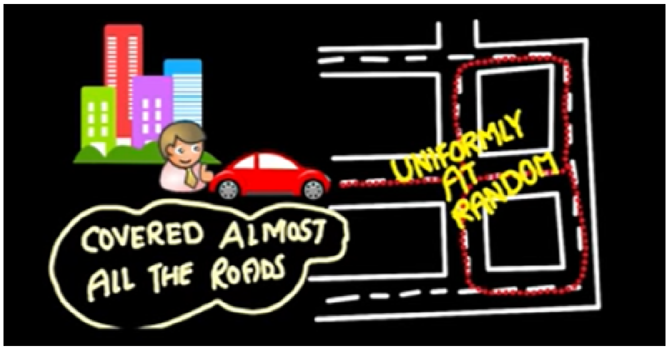
LECTURE 76 - COLLECTING THE WEB GRAPH

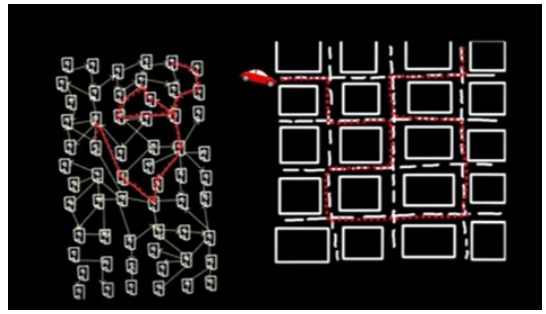
2 ques to tackle - 

Q1 - how to collect?

Imagine - u are put to new city, no one has vehicle, only u have it with a driver, u keep on going randomly at every junction for 1 month.

At this time, u have covered almost all the roads





A Quick Experiment

Random Walk

Generate a Random Graph

Pick a Vertex Randomly

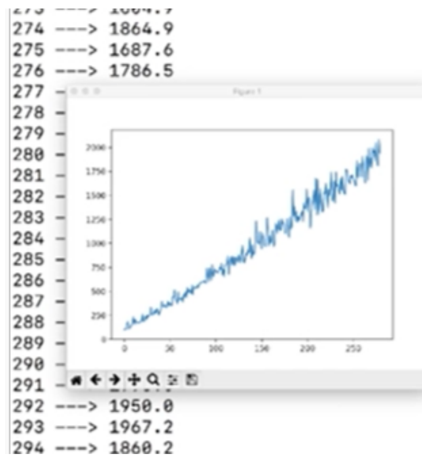
Start Walking on the Graph Randomly

When will one visit all the vertices?

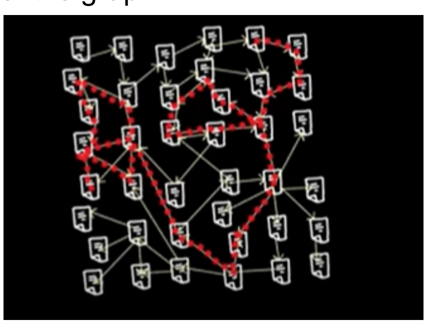
Will one at all visit all the vertices, ever?

Let us write a quick python code to check how many random walks does it take to reach the entire graph...

```python
import random
import matplotlib.pyplot as plt
import networkx as nx
import numpy

def walk(n,p):
    start=random.randint(0,n-1)
    G=nx.erdos_renyi_graph(n,p)
    S=set([])
    v=start
    count=0
    while(len(S)<n):
        Nbr=nx.neighbors(G,v)
        v=random.choice(Nbr)
        S.add(v)
        count=count+1
    return count

l=[]
for i in range(20,300):
    z=[]
    for j in range(10):
        z.append(walk(i,0.3))
    l.append(numpy.average(z))
    print i,"--->",numpy.average(z)
plt.plot(l)
plt.show()
```



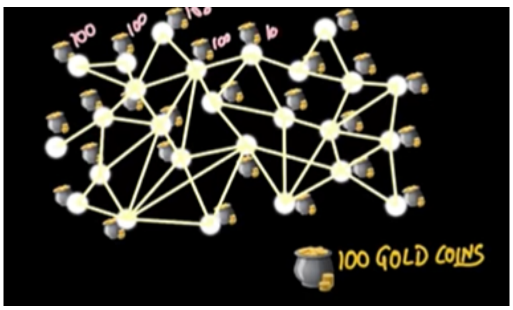
Thus, it is not so difficult to explore all the nodes. U get nodes + edges both = a partial structure of the graph

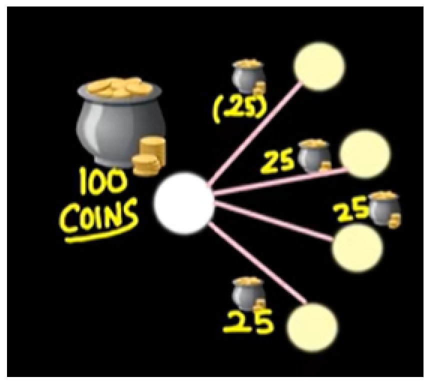


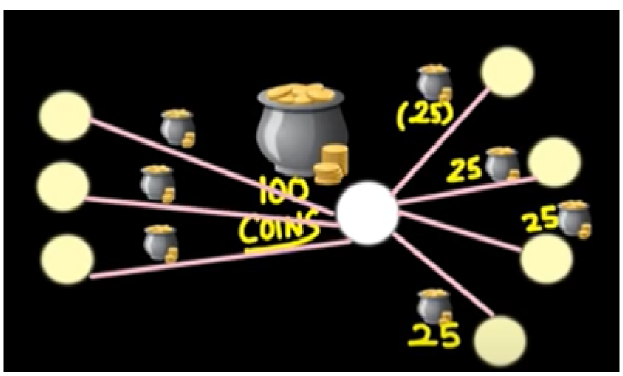
Q2 - how this webgraph solves the problem?

A PUZZLE -

DISTRIBUTE 100 GOLD COINS TO EVERYONE. Every nodes distributes its to its neighbours

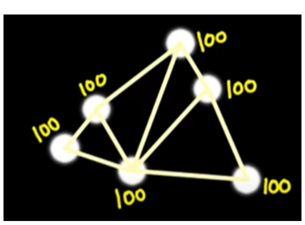


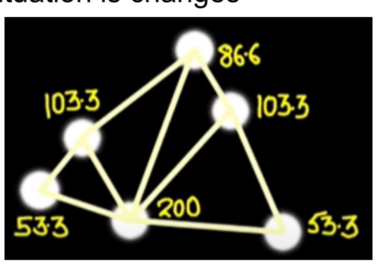


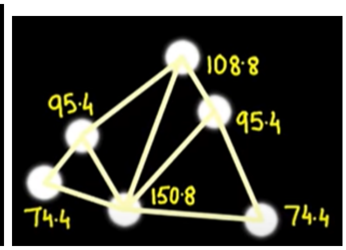


he also gets by others

In every iterations the situation is changes

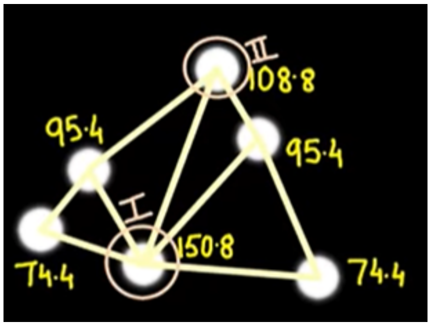






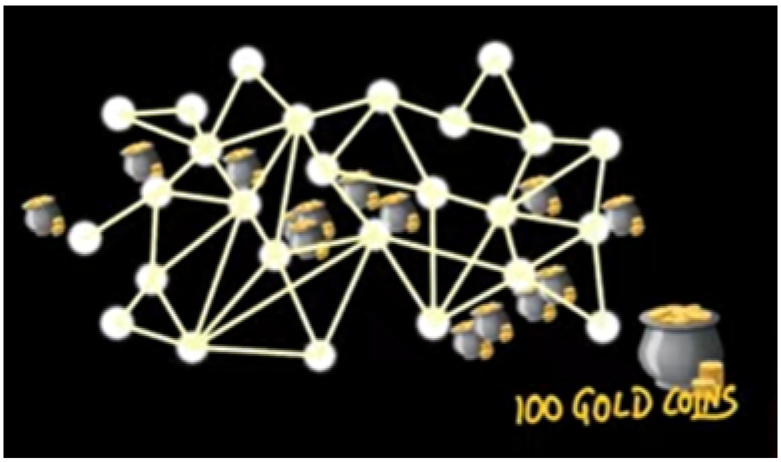
…..

After infinite iterations, this converges. And gold coins dont change. At this stage, u check who had the max gold coins and give him rank 1

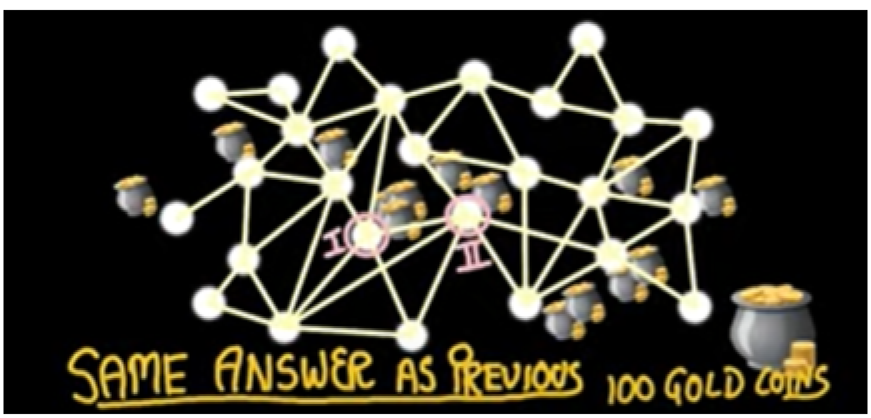


LECTURE 78 - Random Walk Coin Distribution

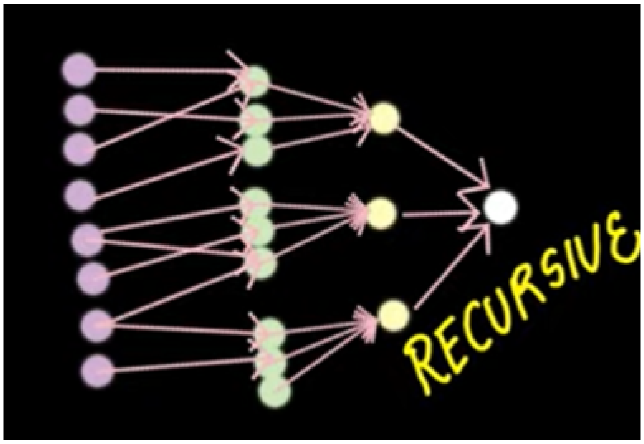
Assume u stand on a node,, u go to a neighbour at random, and give him a gold coin and continue this .

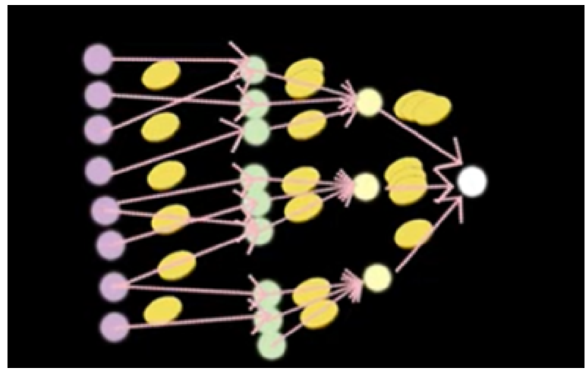


After a million iterations, every node has some gold coins but they arent equal. Check which node has the highest gold coins? We see that it is same as the previous experiment result. Same for rank2 and others too.



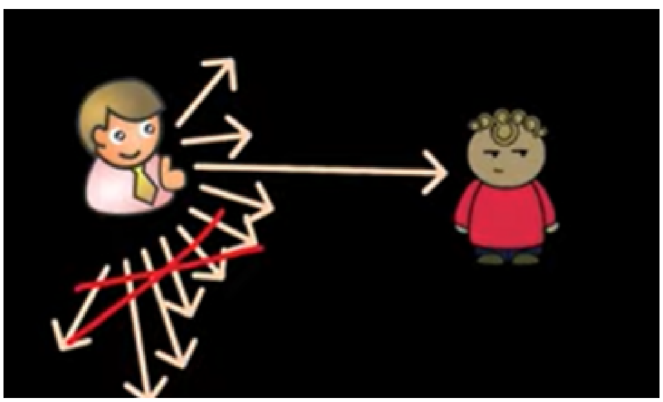
they yield same ranking of the nodes
A node has max coins iff it is getting lot coins from prev nodes, they too are getting lots coins from their previous nodes





Ily, GNU Full form = GNU's Not Unix
2 aspects -

1. Cousin must be very rich to help you
2. He must not have lot of other people to help , else his help is divided and i dont get full.



Lecture 79: Google Page Ranking Using Web Graph

Google crawls the web through all the web pages with the searched keywords, do the coin dropping experiment and displays the pages acc to ranks of pages in terms of gold coins accumulated after sorting it acc to the gold coins

Lecture 80: Implementing PageRank Using Points Distribution Method-1

- Create/take a directed graph with 'n' nodes.
- Assign 100 points to each node.
- Keep distributing points until convergence.
- Get nodes' ranking as per the points accumulated.
- Compare the ranks thus obtained with the ranks obtained from the inbuilt Page Rank method.

STEPS -

Lecture 81, 82, 83 :Implementing PageRank Using Points Distribution = CODE IN VIDEO

Lecture 84: Implementing PageRank Using Random Walk Method -1

- Create/take a directed graph.
- Perform a *random walk*.
- Get sorted nodes as per points accumulated during random walk.
- Compare with the inbuilt Page Rank method.

STEPS -

Lecture 85: Implementing PageRank Using Random Walk Method -2 = Code in video

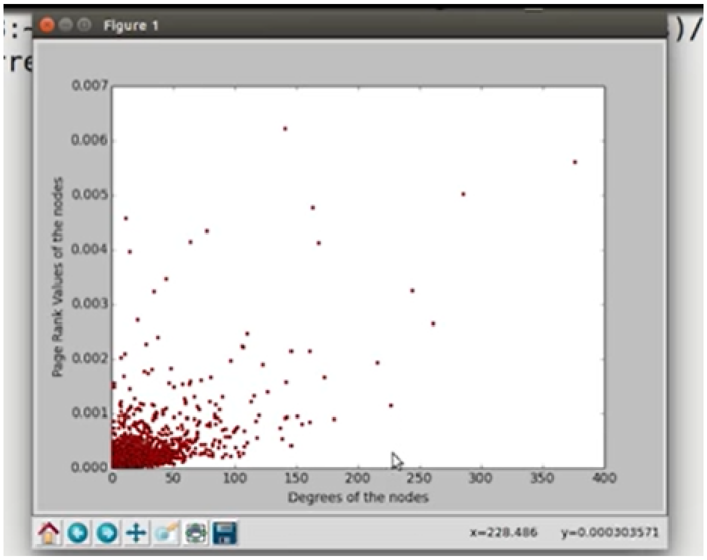
```
nodes_sorted_by_random_walk [1 2 6 9 8 5 3 4 7 0]
1 2 6 9 8 5 3 4 7 0
```

we get same results by both techniques

Lecture 86: DegreeRank versus PageRank

Rank that is we are going to see whether they correlate with each other or not? We are going

to see whether the nodes which have less degree do they have less page rank and vice versa. So,



not linear =>not correlated bcz pages having high indegree might have lower page rank

TELEPORTATION

To avoid the possibility of random walkers getting stuck in vertices with no outgoing edges, we add a teleportation probability , which corresponds to a transition from a vertex to any other vertex , uniformly chosen in .

Teleportation, in the context of random walks on networks, involves introducing a small probability of jumping to any random node in the network, regardless of the current node's connections. This helps prevent the random walk from getting stuck in certain regions of the graph, especially those with low connectivity or dead ends. So, Statement I is true, while Statement II is not entirely accurate; it's more about jumping to any random node in the network with a certain probability, rather than any random node without consideration of the network's structure.

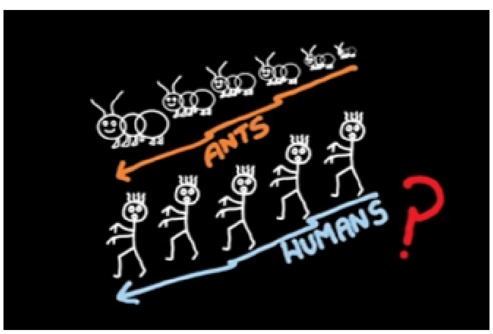
Teleportation consists of connecting each node of the graph to all other nodes. The graph will be then complete. The idea is with a certain probability β, the random walker will jump to another node according to the transition matrix P and with a probability (1-β)/n, it will jump randomly to any node in the graph

A - 4	4	1/16
B - 6	6	6/16 = 3/8
C - 2	2	1/8
D - 4	6	

A 10 20 = 30
B 10 20 10 = 40
C10, 10 10 = 30
D 10 20 = 30
Correct - past yeart

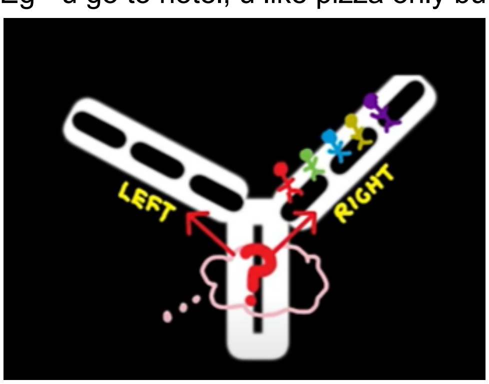
WEEK 7

Lecture 87 - We Follow

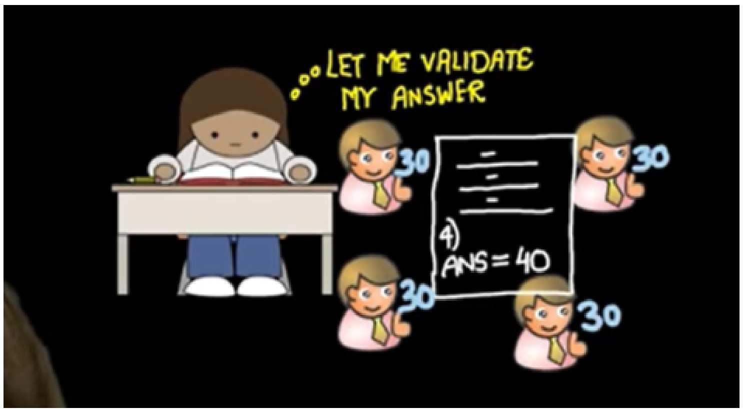


ANTS FOLLOW EACH OTHER

Eg - u go to hotel, u like pizza only but everyone there is eating burger so u tend to take burger



Lecture 88 - Why do we Follow?



3 types of Follow

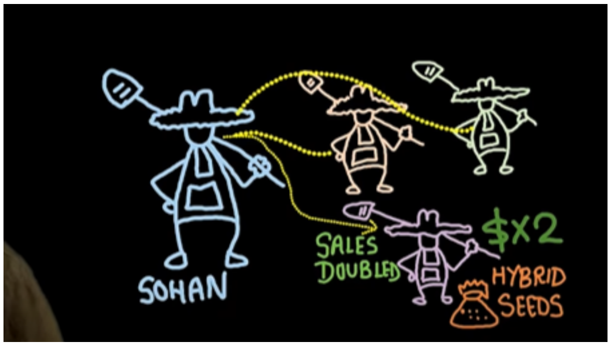
1. Frnd doing something, i start doing it, eg smoking ciggarettes
2. Frnd using a product, i like that and start using it too, eg ipad
3. Information of interest received from a friend, eg info about celebrity, assignment. And i liked and believe this info => i followed him

even if following sometimes lead to bad situations

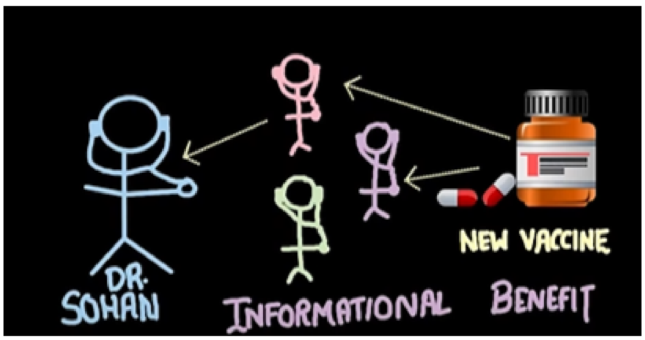
Examples -

1. Frnd is in medical industyr, tells about shoes with new features. And i dont like it tho, but he offers me to give a free pair if i advertise to 5 people.
   So i followed him for some explicit benefit

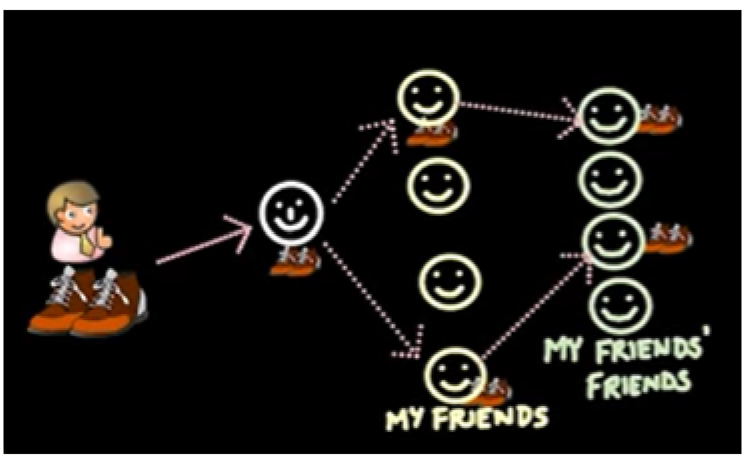
2. Sohan sees that his other farmer friend gets the sales of his crop doubled. On asking, he tells sohan that he bought hybrid seeds imported from america. **Sohan needs to have some friends and follow them to get useful information**



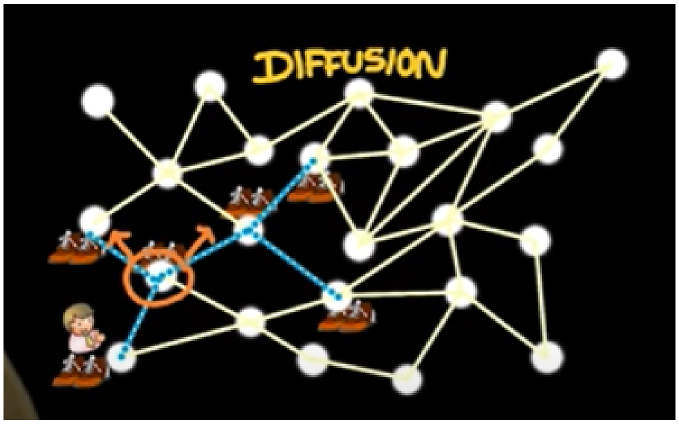
3. Sohan is a doctor. He needs to follow others doctors to get useful info of advances in the medical field. This gain is known as **informational benefit**



Lecture 89 - Diffusion in Networks
I get influenced by the sport shoes, some of my friends will get influenced and further their friends

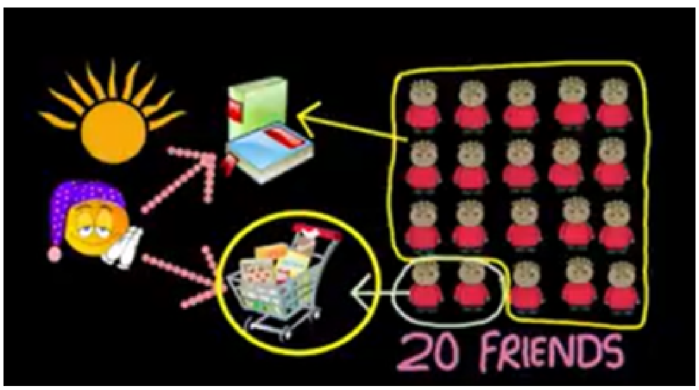


The product is travelling/branching known as diffuison in the network



What will happen in end? Will everyone adopt it? Or die in between? - it totally depends on what product or info is getting diffused in the network

Lecture 90 - Modeling Diffusion



you might choose to study in library rather than having fun

Factors acc to which we choose

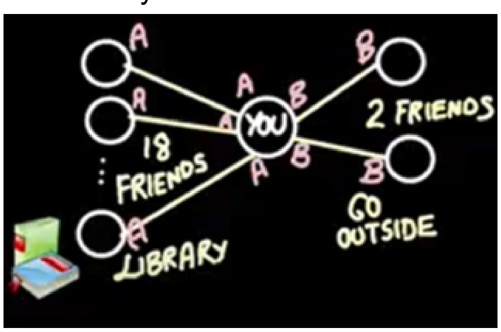
1. PAYOFF

when i think of it alone

depends on what your friends think

- **2 people** get a payoff associated with an activity when they do a common thing together

But in real you have alot of friends - **20 friends**



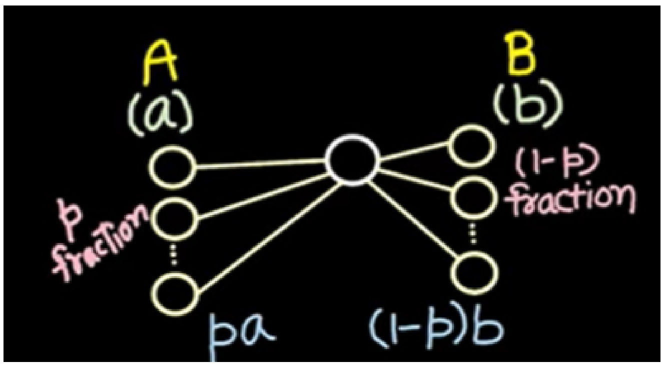
to choose, we calculate and compare total payoff of both the situationa

in this situation, u choose to sit in library

here u choose to go out and have fun.

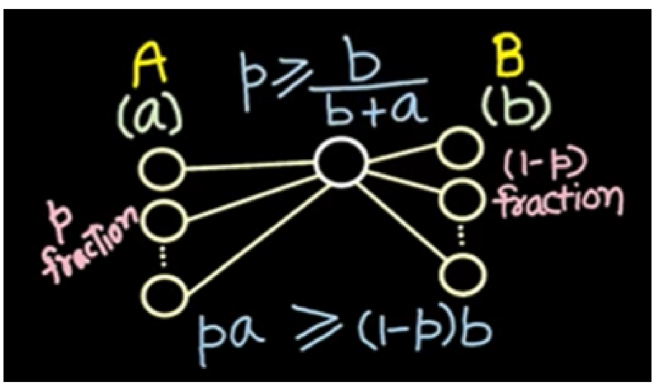
Lecture 91- Modeling Diffusion (Continued)

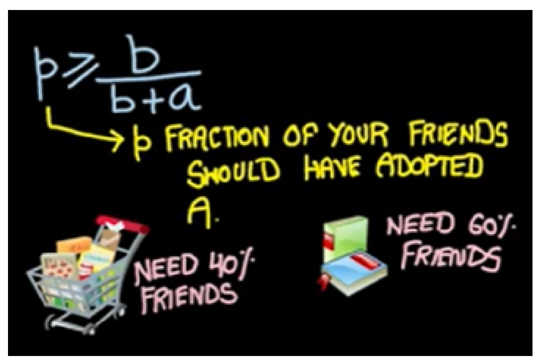
2 choices - A and B having payoff a and b resp



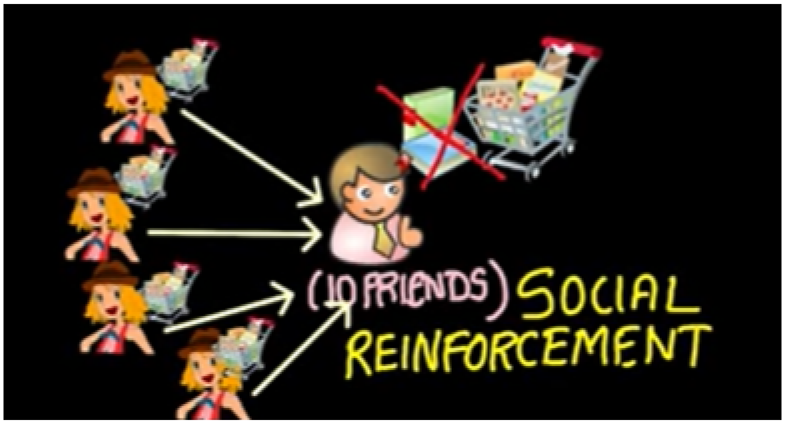
p = fraction of frnds

For A to be adopted -





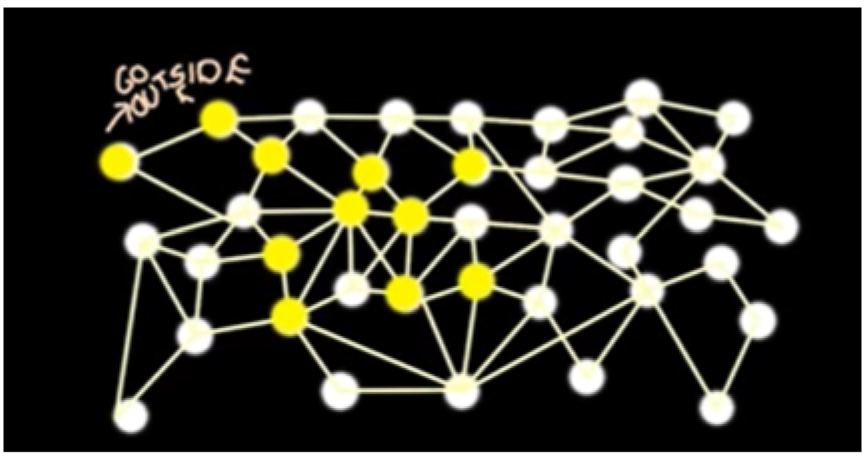
P is the threshold value which is needed to decide what to choose



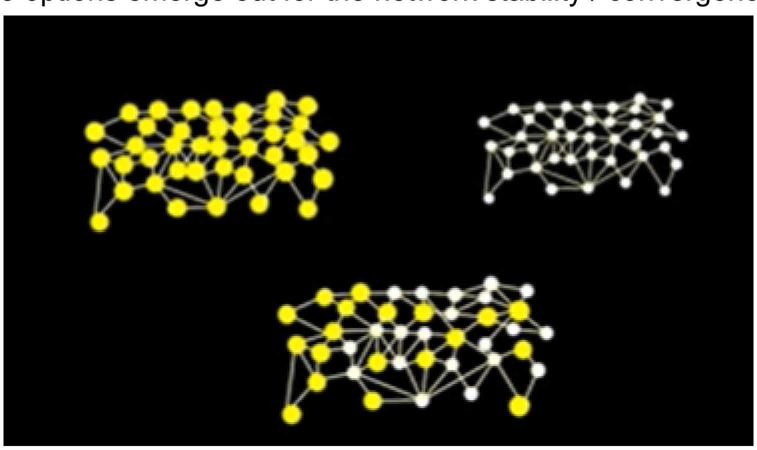
when 4 friends came , i changed my mind of studying

This is known as **SOCIAL REINFORCEMENT** - when more and more people come and tell something, we tend to believe this.

Eg - first entire class thought of studying but 2 people flipped now others also started following them

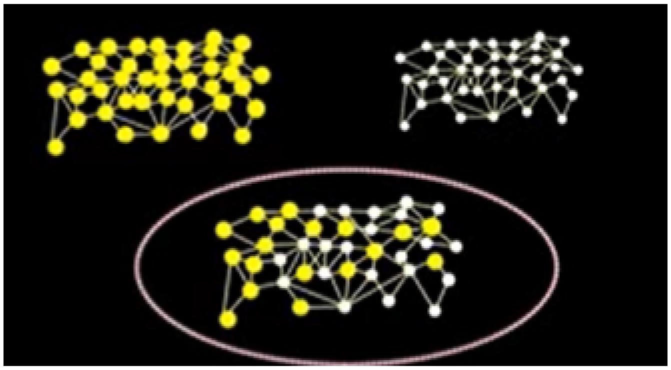


3 options emerge out for the network stability / convergence -

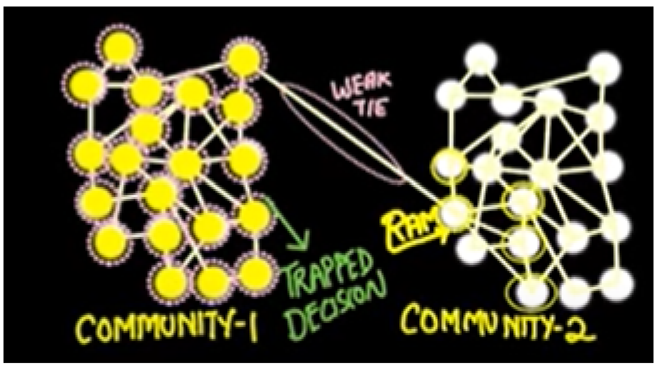


1. All go out
2. All flip back again and study
3. 2 parties are generated in the class

Lecture 92 - Impact of Communities on Diffusion



Class splits into 2 parties - when there are weak ties between 2 strong communities



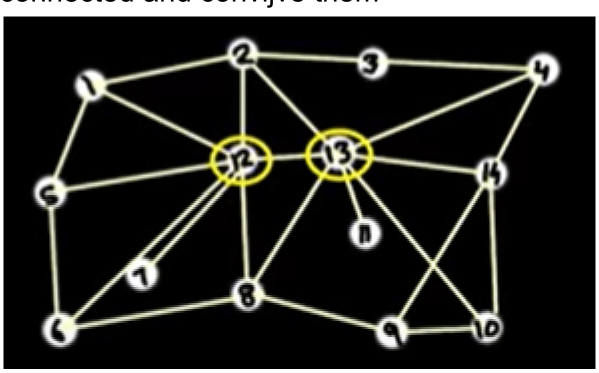
com1 cannot easily change ram decision . this is how 2 coms are formed, the decision is not easily passed from one com to the other

PROBLEMS with diffusion

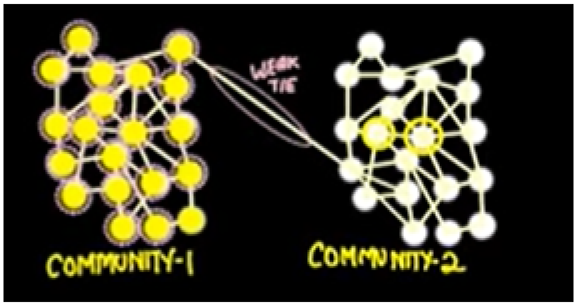
1. STARTING TROUBLE - People dont tend to take risk for the first time to start / adopt something new

SOLUTION : 1. Increase the payoff so that people become interested to adapt it - not so fruitful

2. Focus on KEY PEOPLE
   Eg - if u need a bike, choose to convince family head. Or use big celebrity SRK for advertising . this is knwn as VIRAL MARKETING. Identify the key nodes which re well connected and convijve them

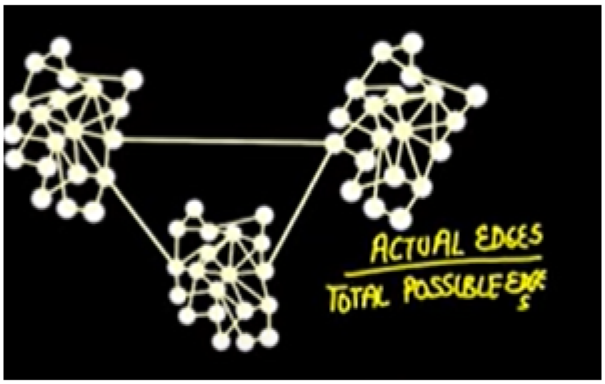


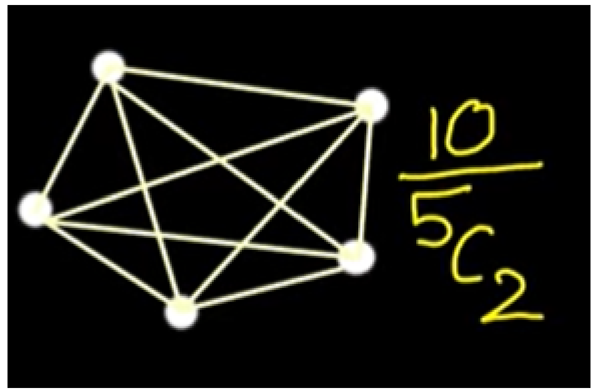
12 , 13 are key people



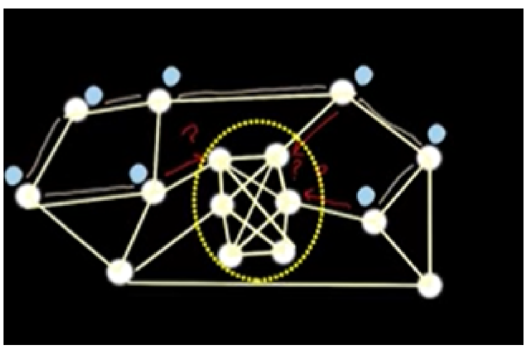
convince these 2 key people

DENSITY OF COMMUNITY - how well connected a community is.



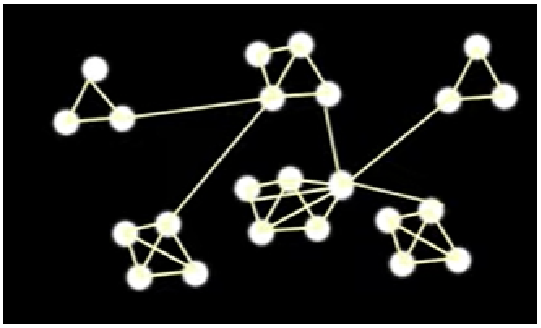


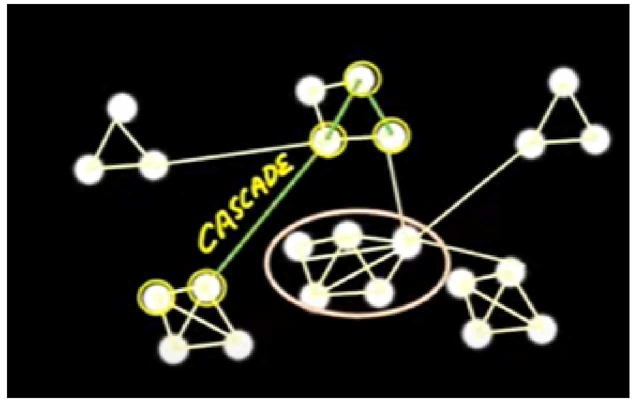
Is density related to whether info will be able to spread in the network or not?


a strong dense community is difficult to convince

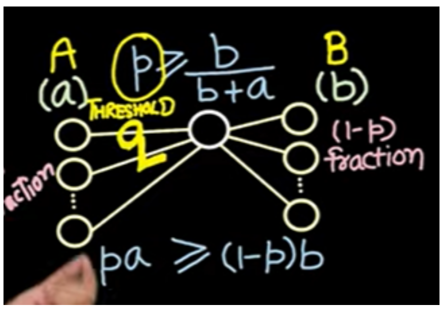
Higher the density , difficult is the idea of diffusion of info in the network

Lecture 93 - Cascade and Clusters



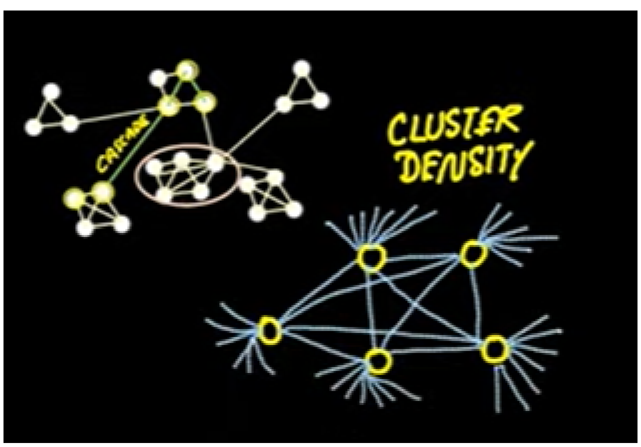


We will see whether it becomes a complete cascade or not - whether everyone in whole class will be convinced or not?



assume all nodes have same threshold q

Density of a cluster = D if if you look at every node in this cluster and at least D fraction of these nodes friends is in this cluster itself.

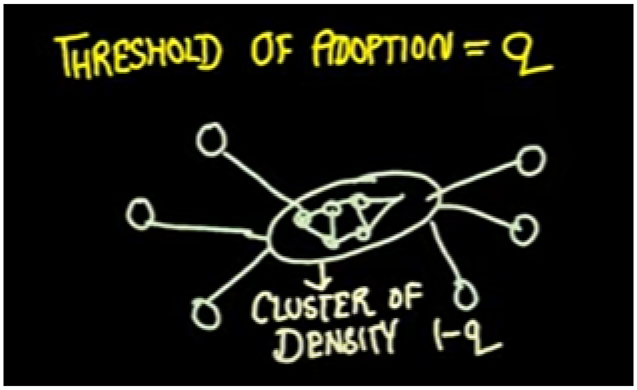


in the cluster , if everyone has 10 friends, 3 should be inside this particular cluster to make cluster density = 0.3

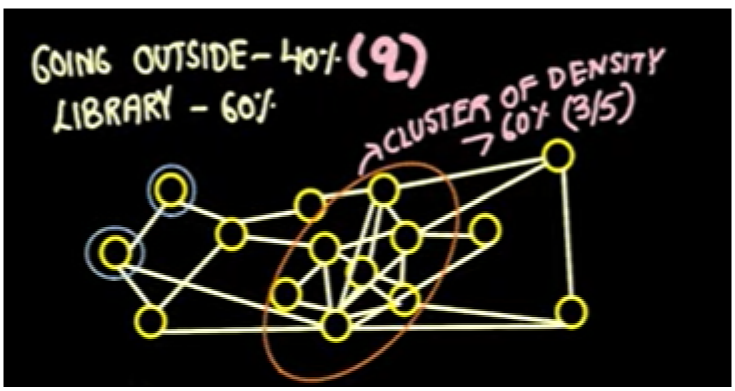
CLAIM - the threshold of adoption for every person is q as we said before. Then the cascade cannot complete itself, complete itself means starting from some nodes it cannot sweep the entire network,

the cascade cannot complete itself. If there exist a cluster in this network

of density greater than 1 minus q.



EXAMPLE -



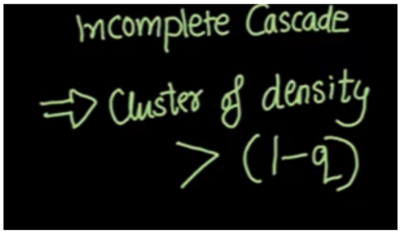
when idea tries to enter 1-q density cluster, the threshold condition is not achieved.
Thus, cascade cannot enter the cluster

CLAIM 2 -
If a cascade couldnot complete itself, =>

happens that the cascade is not complete. This implies that this shows that there is

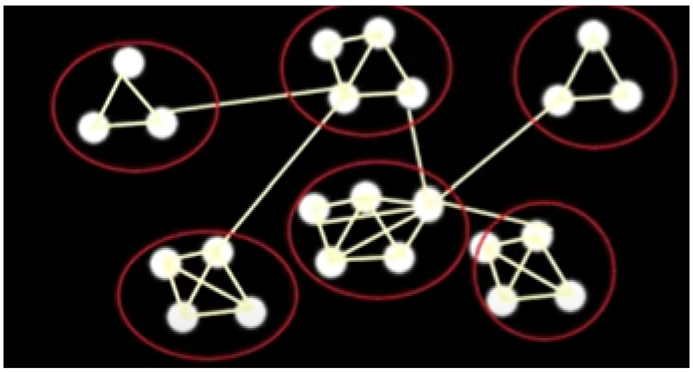
a cluster of density greater than 1 minus q in the network. So, it is a by implication,

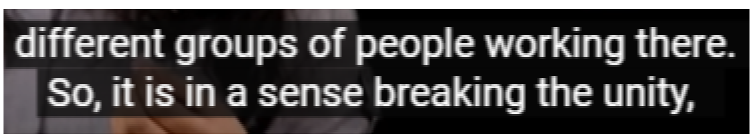
- both ways

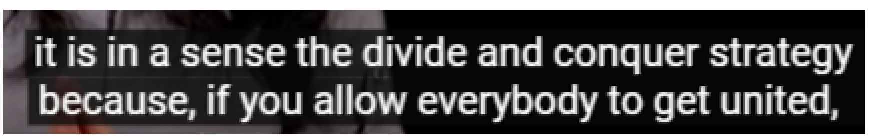


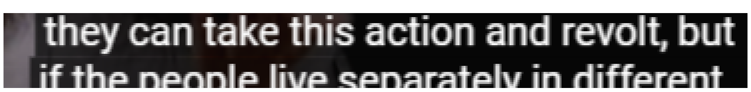
Lecture 94 - Knowledge, Thresholds and the Collective Action

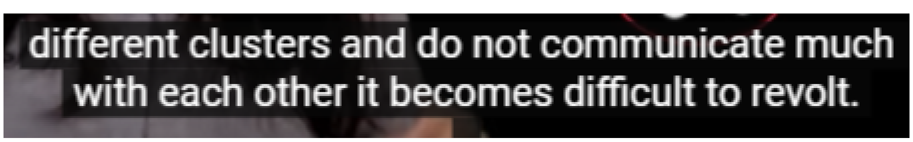
The clusters are not inter connected and thus not ready to revolt together with others against something wrong.



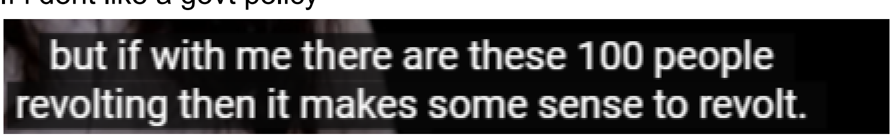








If i dont like a govt policy -



So, what I will be doing is I will be looking
at these other people whether they areready to

revolt or not. And if I am suspicious that yes
these people are not going to unite with me I

will also not go and revolt. Same is the situation
of let us say people want to put some people want

This scenario is known as collective action.

HOW DO WE MODEL THIS COLLECTIVE ACTION ?

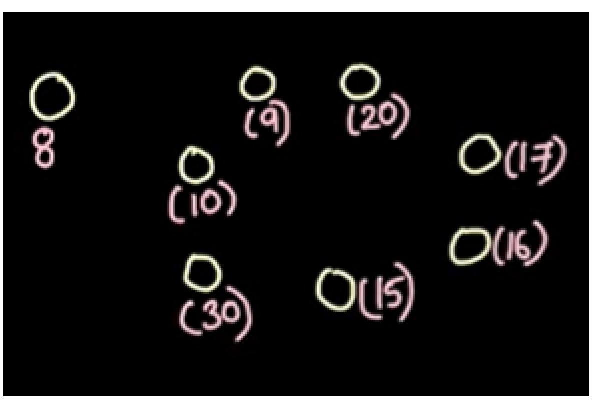
1. ASSUME that every person has an intrinsic threshold.

   ready to revolt. So, 7 others total 8, 1 is me 7
   others, so if 7 more people are ready to revolt,

   I have an intrinsic threshold of 8. So, it means
   that if including me, I say that if 8 persons are

   I will go ahead and revolt. And then there will
   be a threshold associated with every person.

   So, each of the person will have a different,
   different threshold. Somebody will say I need.



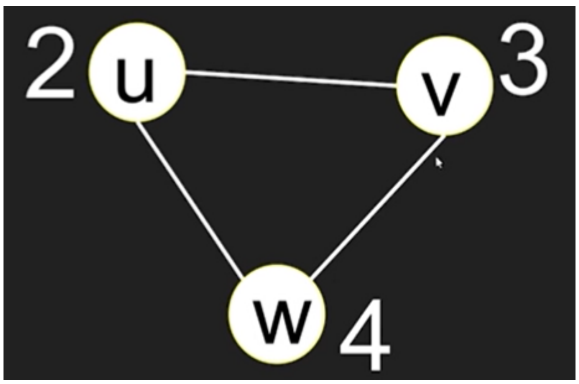
2. threshold. Now how do we decide what to do. So, to decide what to do, here we kind of know what

is the threshold of our close friends. So, if I have a friend, let us say Mira,

So, here I have an idea of my threshold, I have an idea of my friends is threshold and based on

just these two ideas I have to decide whether to revolt or not. So, in the next screen cast

SCREENCAST - whether there will be protest or not?



U needs 2 people to participate in the pro test. So, 1 person 2 imputing itself, so 1 obviously

will be node U and then it needs 1 more person to participate in the pro test for it to participate

the threshold for node W is 4 and W has only 2 friends right. So, even if both of these

friends participate in the pro test, W cannot participate because, it needs at least 4 people,

so W drops the idea of participating. Let us look at node V. It needs 3 people. So, node V looks at W and then it knows that W cannot participate in the protests because,

V now knows that W is are going to participate. So, again its threshold will not be reached and

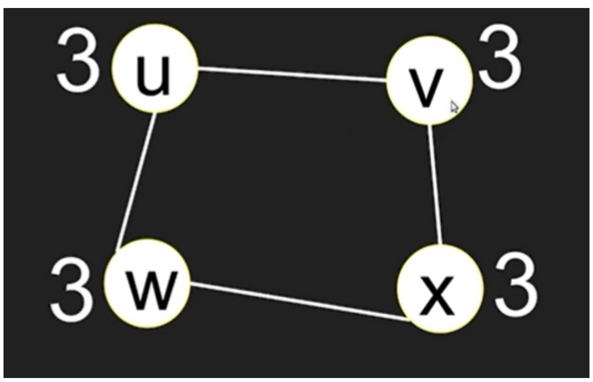
node V drops the idea to participate. Node U only requires 2 people right? Only requires 2

people to participate, but then node U it know the thresholds of both of these nodes right.

And then hence it can determine that none of these people neither V nor W is going to participate.

Hence node U also decides to drop the idea of protesting. Hence the protest does not happen

EXAMPLE - 2



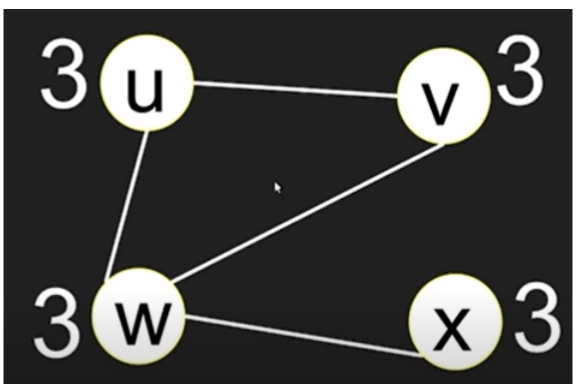
But if we look at this case, it seems quiteeasy

for a protest to happen. So, any 3 nodes along this cycle can get together and do a protest. U,

V and W if they get together they can do a protest, W, X and V can do a protest

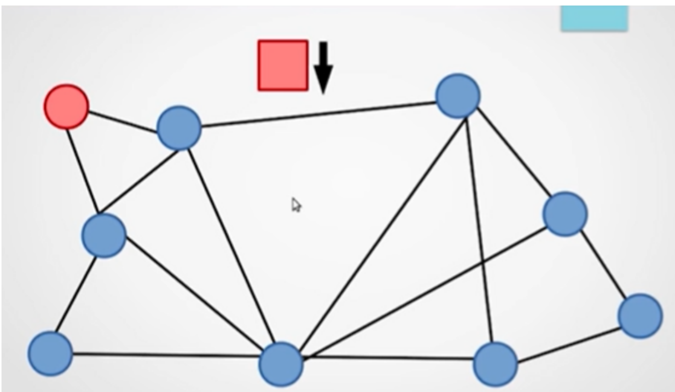
and so on right, U, V and X can do a protest. But then every node has some incomplete knowledge, they cannot see the entire network. For example, node U here have no idea about node X right. So, node U cannot see this entire network and then come out with the conclusion that yes, it can go get together with with V and W and protest. So, let us see how does it happen at thelevel of every individual node. So, if we look at node U. Node u knows that node V and node W they have a threshold of 3 right. And hence they can get together and do a protest, but then the problem is node U also knows that see, V and W both are the friends of node U and then node U knows that V and W are node friends. They do not know about each others thresholds. Node U knows that W has a threshold of 3 V has a threshold of 3, but then node V and W have no information about each other right. So, we do not know what they will be doing and then node U does not have any idea about the threshold of node X. So, node U can think that may be the threshold of node X is very high let us say 5. So, if the threshold of node X is very high, here it is 5 then what will happen? V might decide node to protest because, X will not be protesting because, it has a very high threshold, then V will not be protesting and then how can you know this node U protest here right. So, it is unsafe for you to participate in the protest and it drops its idea of participating. And the same thing happens with all of these three nodes V, X and W because, V does not know about W's threshold, X does not know about U's threshold, W does not know about V's threshold. So, even if in this case actually they can get together and do a protest, but then when these people look at the networks from their perspective they drop the idea of protesting.

EXAMPLE 3 -

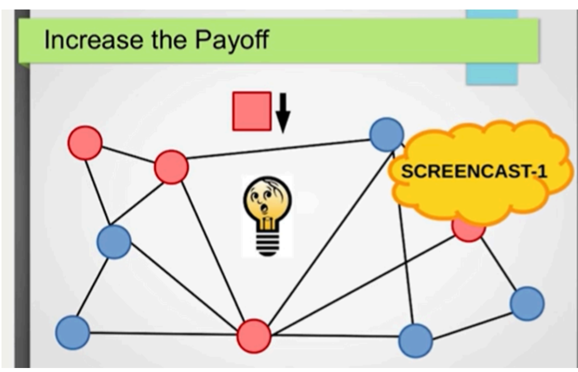


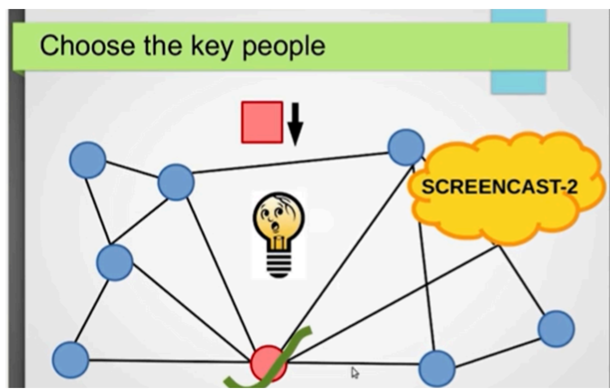
can very easily happen here. Can happen here and also these people you can see that a protest can happen here. You can look at V and W and then look at their thresholds. And then here you can be sure that yes V and W will participate because, V and W also knows about each other's threshold. So, here mainly these 3 people U, V and W can get together and go for a protest.

Lecture 95 - An Introduction to the Programming Screencast (Coding 4 major ideas)
4 main concepts -

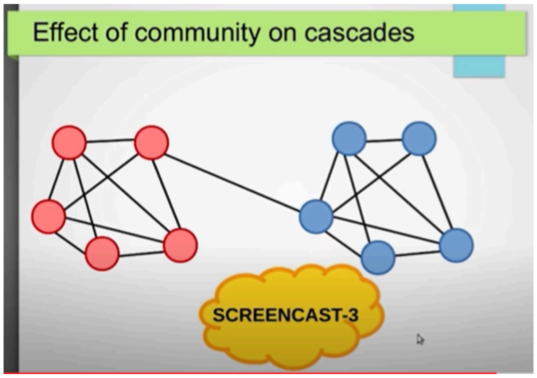


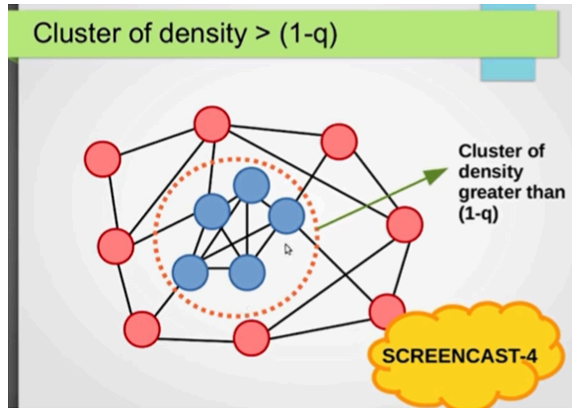
new red idea is introduced but people feel risky to adopt this , 2 soluti-





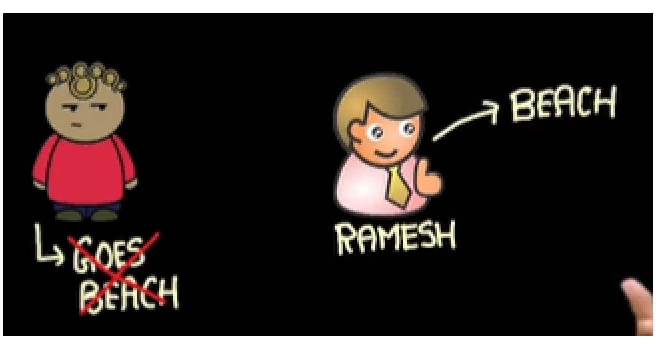
—--



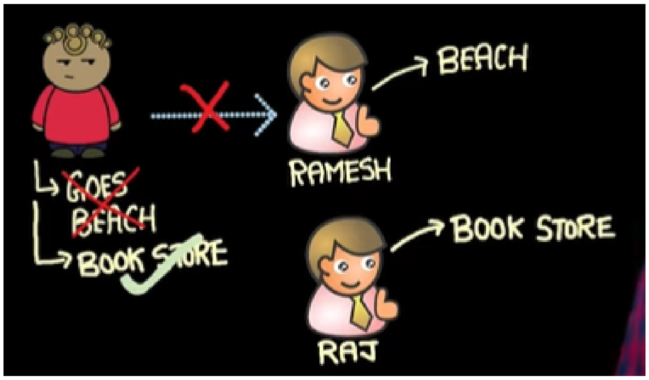


WEEK 8

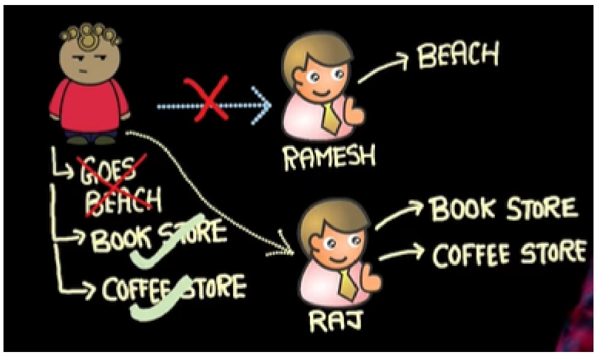
Lecture 101 : Introduction to Hubs and Authorities (A Story)



FRND suggested to explore beach, i went but didnt like it much. Now next day, i wont ask ramesh, bcz he may give similar suggestion,. Ask new frnd raj



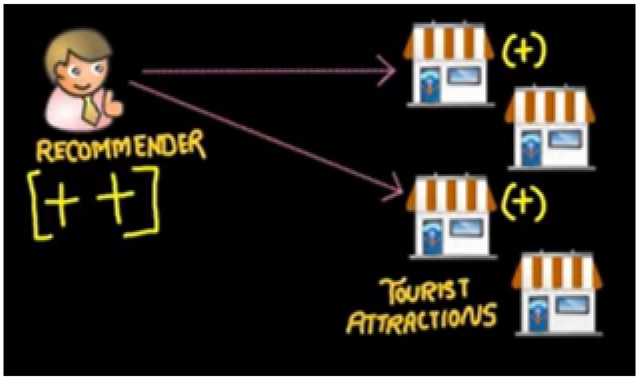
nextday i again ask him for suggestion

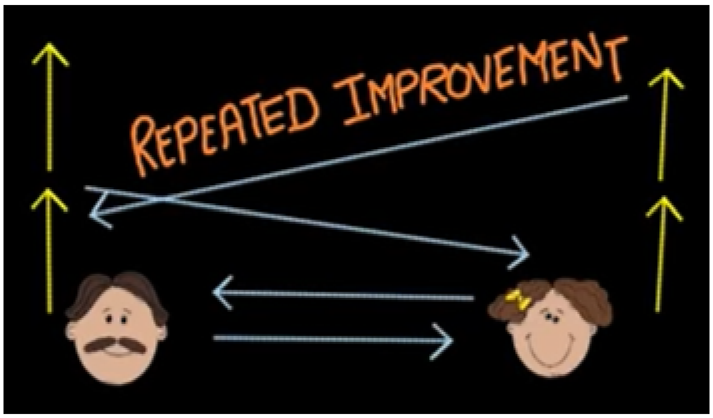


i enjoyed alot again

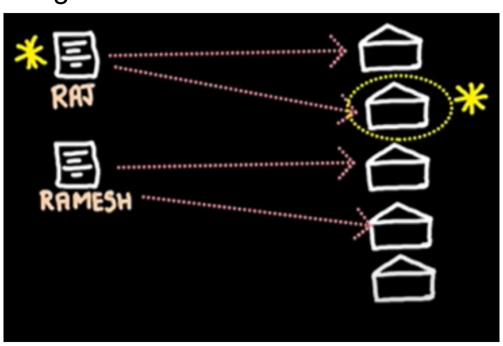
When u visit a place, u give it some stars, but the person who recommended this is also rated. And more is the rating of person more is his opinion valued

Lecture 102: Principle of Repeated Improvement (A story)





u are happy, u keep spouse happy and this go on



raj and ramesh are list of placesvisited . now people visit the place, give place rating and then give good rating to the list

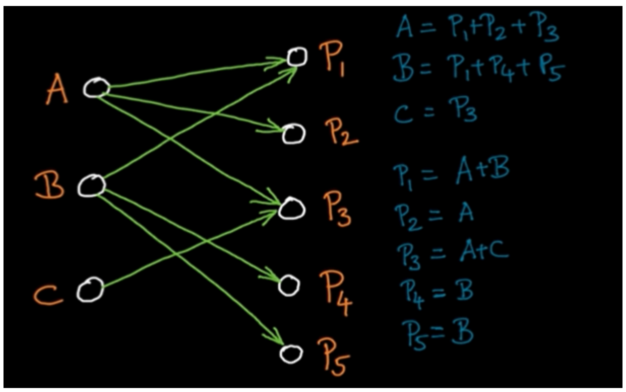
. If the resource is good,

the pointer get some credit, if the pointer has good credit, whatever he points to;

also is a good resource, and resource gets good credit to; is all I am that I am saying. What

Lecture 103: Principle of Repeated Improvement (An example)

Let me consider; 3 people on the lab, and five places, on the right; place 1, place 2 and so on.



fx =A3/(A3+B3+C3)

	A	B	C	D	E	F	G	H
1	A	B	C	P1	P2	P3	P4	P5
2	1	1	1	1	1	1	1	1
3	3	3	1	2	1	2	1	1
4	0.4285714286	0.4285714286	0.1428571429	0.2857142857	0.1428571429	0.2857142857	0.1428571429	0.1428571429
5								
6								

Row 4 has all normalized values.
Keep on repeating the addition and then normalising
2nd iteration

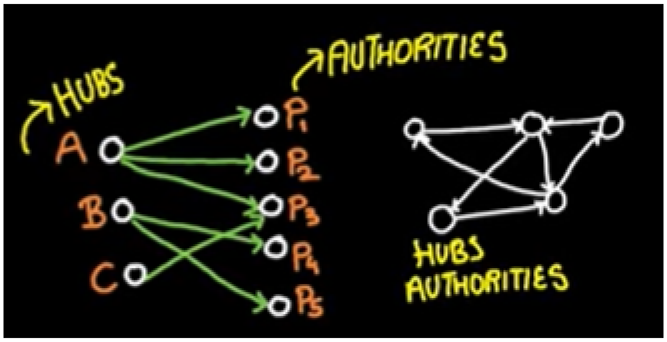
=A5/(A5+B5+C5)

A	B	C	D	E	F	G	H
A	B	C	P1	P2	P3	P4	P5
1	1	1	1	1	1	1	1
3	3	1	2	1	2	1	1
0.4285714286	0.4285714286	0.1428571429	0.2857142857	0.1428571429	0.2857142857	0.1428571429	0.1428571429
0.7142857143	0.5714285714	0.2857142857	0.8571428571	0.4285714286	0.5714285714	0.4285714286	0.4285714286
0.4545454545	0.3636363636	0.1818181818	0.3157894737	0.1578947368	0.2105263158	0.1578947368	0.1578947368

After few 200 iterations,
Values start to converge

197	0.7092753594	0.6061734475	0.2237400659	0.8546376797	0.4608111272	0.6061734475	0.3938265525	0.3938265525
198	**0.4608111272**	**0.3938265525**	**0.1453623203**	**0.3154488069**	**0.1700864866**	**0.2237400659**	**0.1453623203**	**0.1453623203**
199	0.7092753594	0.6061734475	0.2237400659	0.8546376797	0.4608111272	0.6061734475	0.3938265525	0.3938265525
200	**0.4608111272**	**0.3938265525**	**0.1453623203**	**0.3154488069**	**0.1700864866**	**0.2237400659**	**0.1453623203**	**0.1453623203**
201								
202								

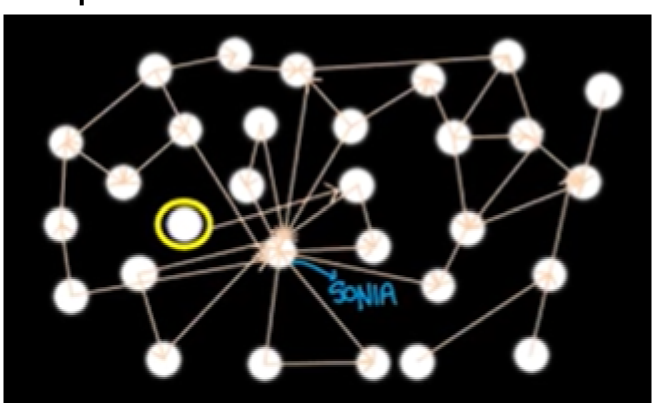
Lecture 104 : Hubs and Authorities
Places - authorities, recommender - hub



. So, given a graph G, all I am saying is;

a node can be given points as hubs. It
can even be given points as authorities;

Example -



people pointing to each other who can solve your problem

Every person has 2 parameters -

That given a node,

you try to give both the values stored.
How well the node is being pointed to is

one value to the node; how well is the node
pointing to is another value of the node.

that definition itself is slightly recursive, you
see a node is good if it is pointing to people;

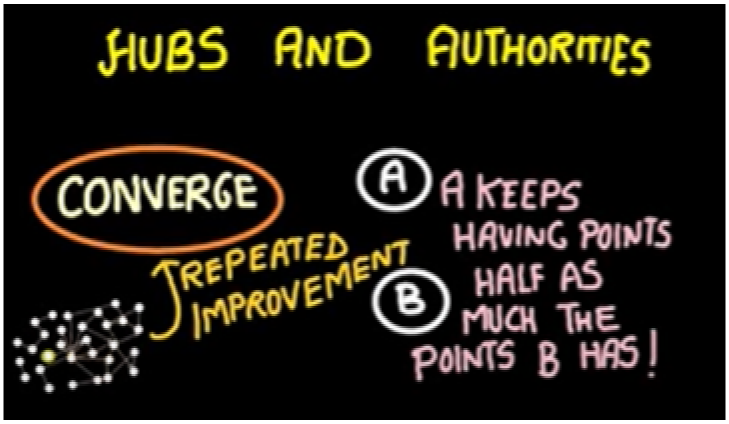
and people are good, if they are being pointed
at by good people. And how do you achieve this,

you achieve this by repeated improvement.
where are our earth is this useful .

Ramesh example I gave you right. So, this was
the way in which you could rate a bunch of bolts,

which would point you to nice news articles.
And that was a birth of this concept, called,

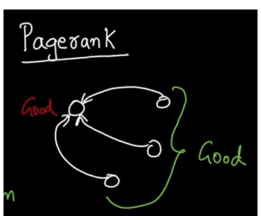
called hubs and authorities



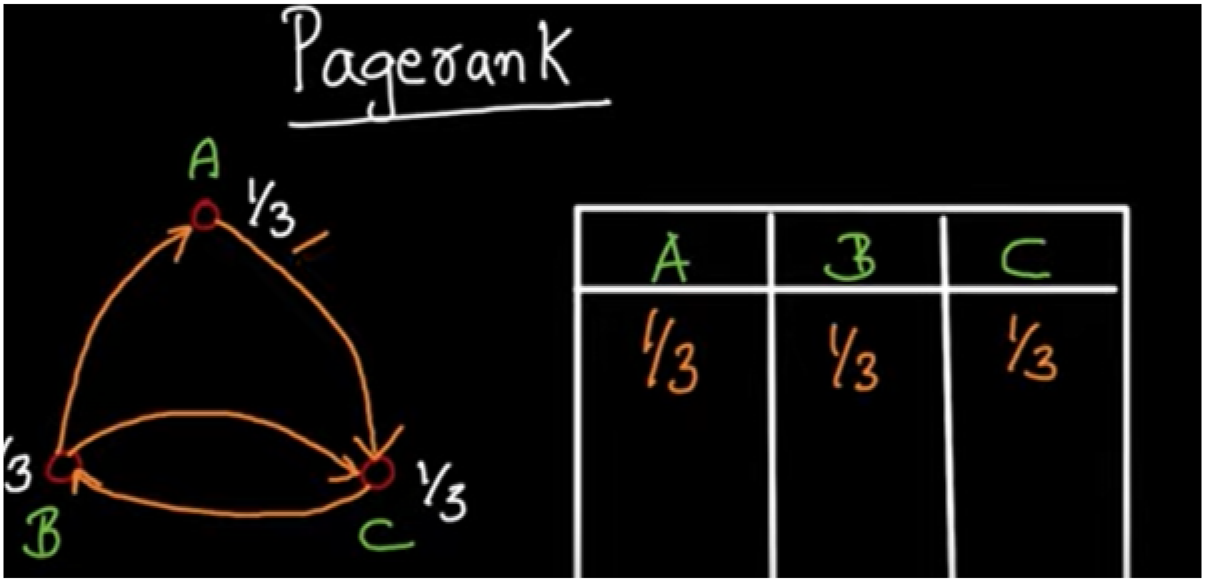
graph converges after few iterations

Lecture 105 : PageRank Revisited - An example

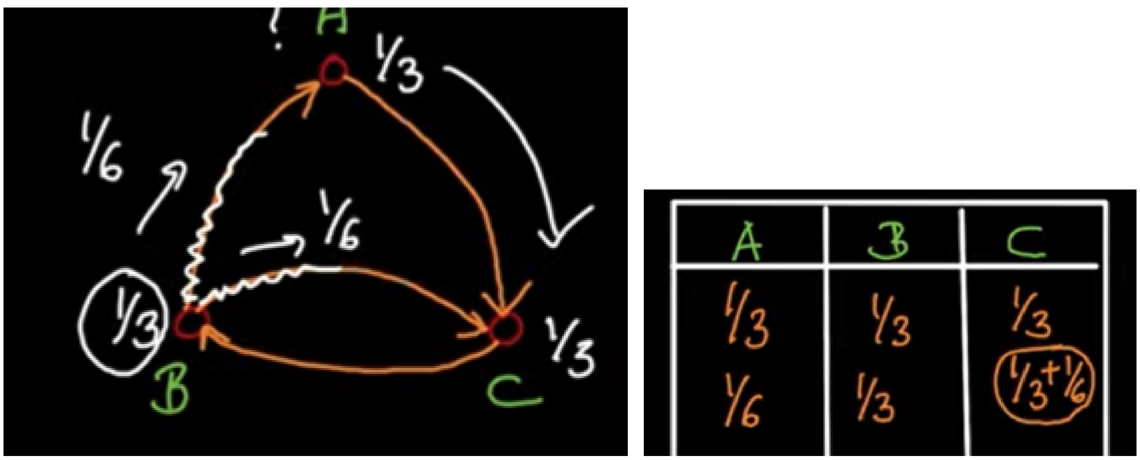
PAGERANK - it is all about hm what you are is decided by who your friends are. If good people point to you it means that you are good. So, you are good if all these people who are pointing to you are good.



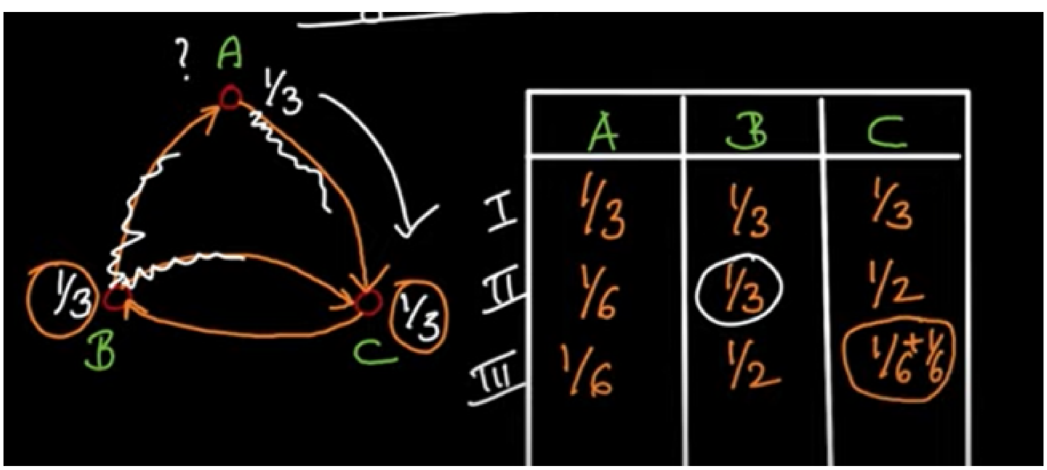
Assume - initially all the 3 nodes have equal resources - ⅓



2nd iteration -
C gets complete A, b gets complete c



3rd iteration-

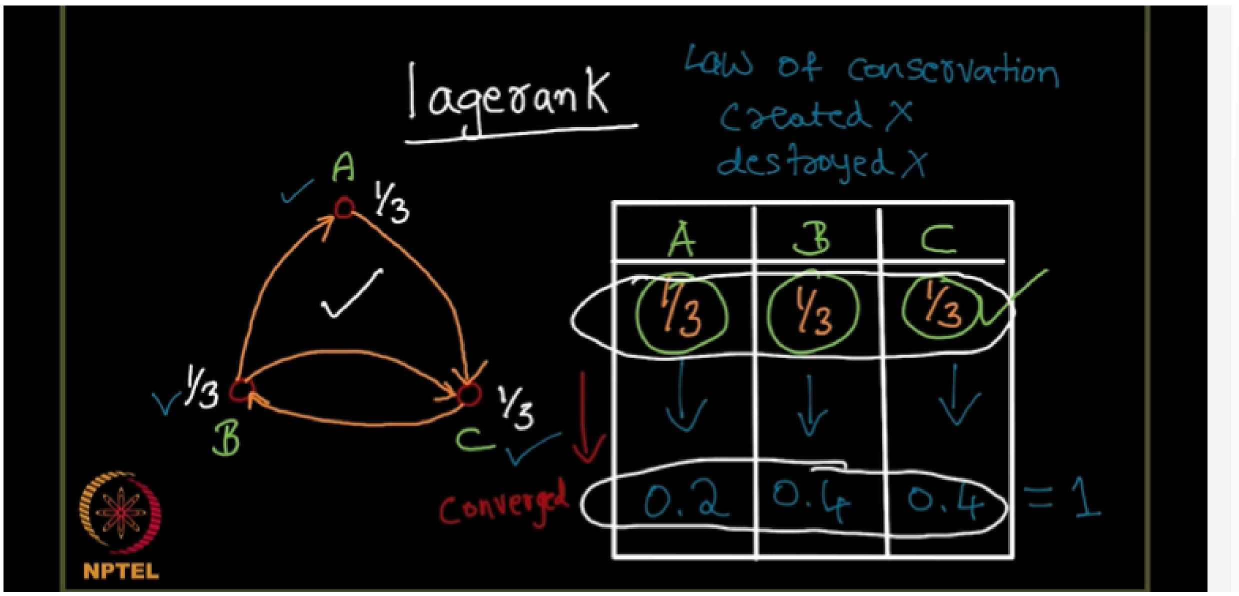


If these goes on… will it converge?

Lecture 106: PageRank Revisited - Convergence in the Example

1	0.1999999999	0.4000000004	0.3999999998	
2	0.2000000002	0.3999999998	0.4000000001	
3	0.1999999999	0.4000000001	0.4000000001	
4	0.2	0.4000000001	0.3999999999	
5	0.2	0.3999999999	0.4000000001	
6	0.2	0.4000000001	0.4	
7	0.2	0.4	0.4	CONVERGES!
8	0.2	0.4	0.4	

Lecture 107 : PageRank Revisited - Conservation and Convergence

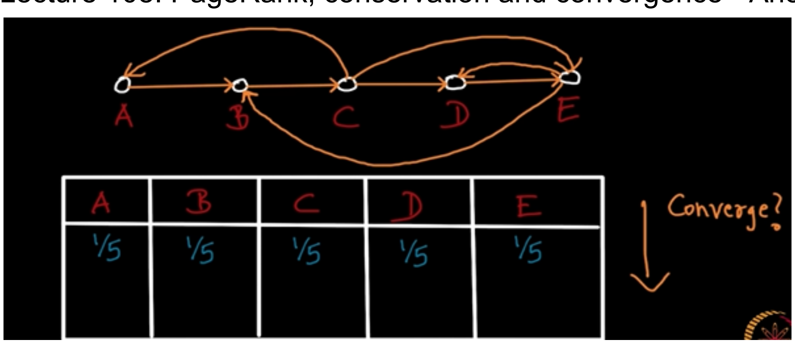


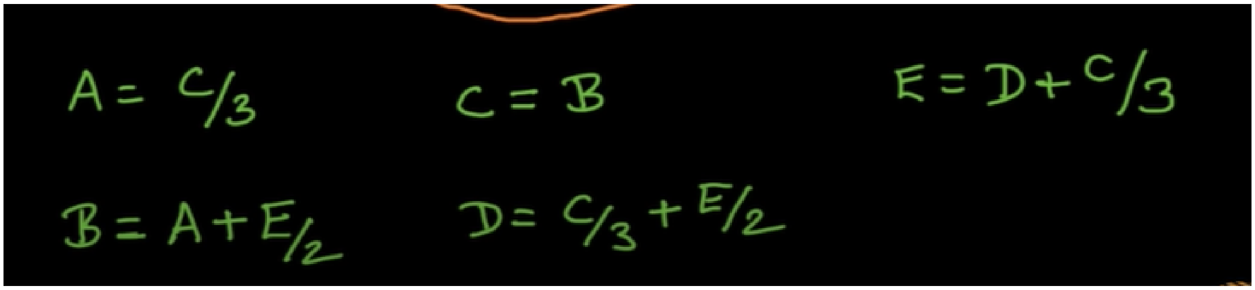
why is that? That is very easy to see the resources that you have distributed to these

nodes do not go out anywhere, they just remain in the network itself. This is like the law of conservation of energy

MAYBE THE CONVERGENCE HERE IS COINCIDENTAL - MAYBE IT DOESNOT CONVERGE FOR SOME OTHER COMPLEX GRAPH

Lecture 108: PageRank, conservation and convergence - Another example



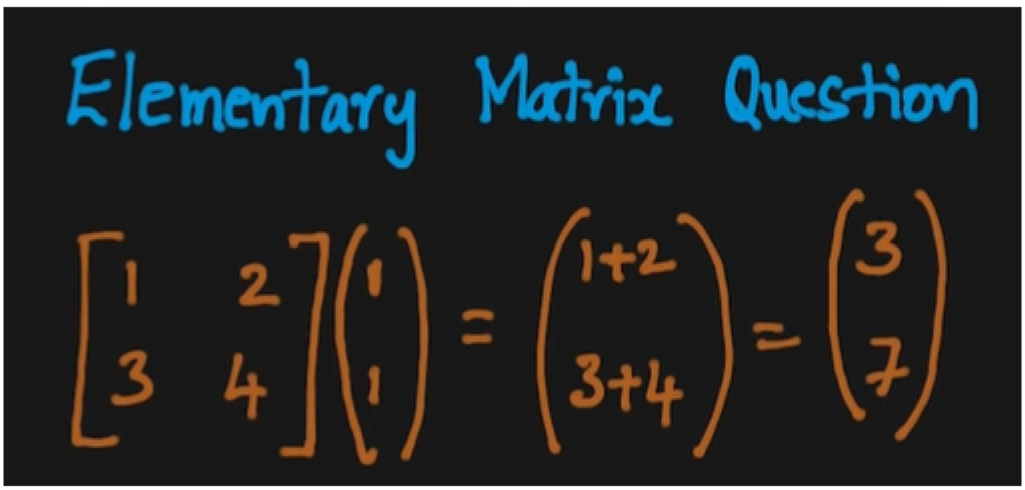


ON ADDING ALL THE RHS, you should get A+B+C+D+E

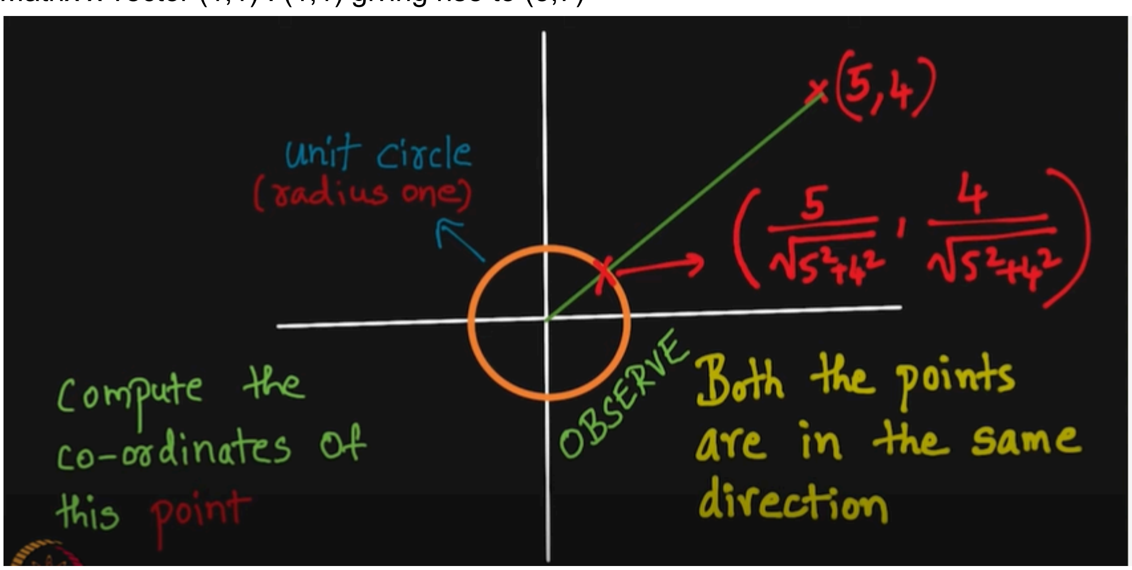
0.07142857143	0.2142857143	0.2142857143	0.2142857143	0.2857142857
0.07142857143	0.2142857143	0.2142857143	0.2142857143	0.2857142857
0.07142857143	0.2142857143	0.2142857143	0.2142857143	0.2857142857
0.07142857143	0.2142857143	0.2142857143	0.2142857143	0.2857142857

YES IT CONVERGES !!

Lecture 109 : Matrix Multiplication (Pre-requisite 1)



Matrix x vector (1,1) . (1,1) giving rise to (3,7)



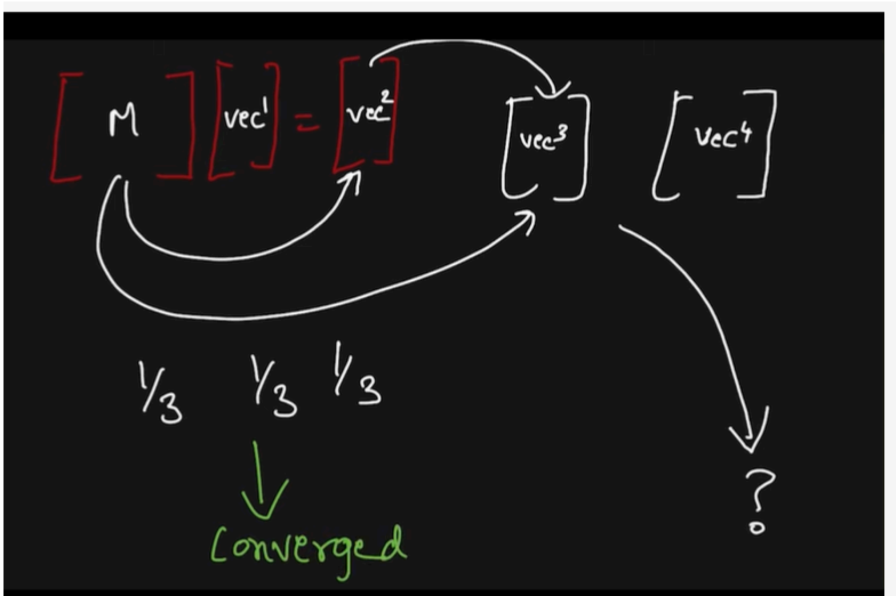
$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{pmatrix}1\\1\end{pmatrix}=\begin{pmatrix}1+2\\3+4\end{pmatrix}=\begin{pmatrix}3\\7\end{pmatrix}$$

$$\begin{pmatrix}3\\7\end{pmatrix}\longrightarrow\begin{pmatrix}3/\sqrt{3^2+7^2}\\7/\sqrt{3^2+7^2}\end{pmatrix}=\begin{pmatrix}0.39\\0.91\end{pmatrix}$$

Normalizing (3,7) to get (0.39,0.91)

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{pmatrix}0.39\\0.91\end{pmatrix}=\begin{pmatrix}2.21\\4.81\end{pmatrix}$$

$$\begin{pmatrix}2.21\\4.81\end{pmatrix}\longrightarrow \text{normalise \& repeat this process.}$$



Matrix multiplication in python

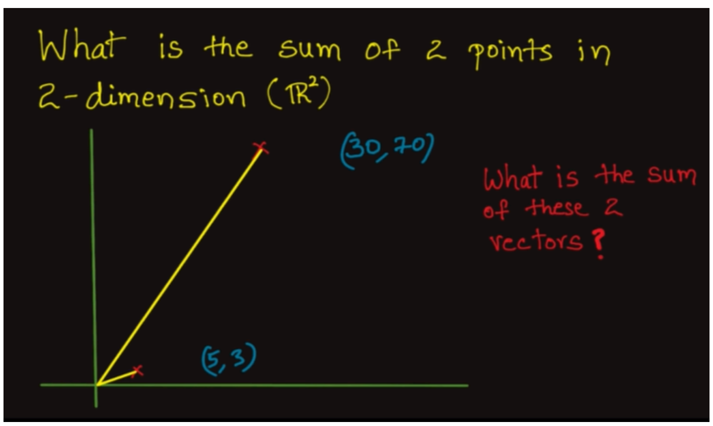
```
In [1]: import numpy

In [2]: A=numpy.mat('1 2;3 4')

In [3]: v=numpy.mat('1;1')

In [4]: A*v
Out[4]:
matrix([[3],
        [7]])
```

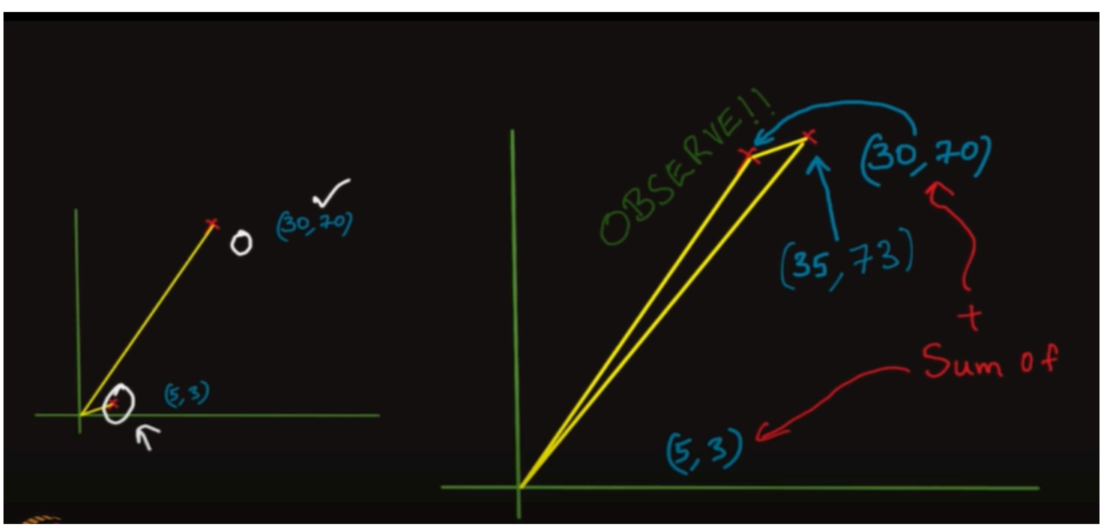
```python
import numpy

A=numpy.mat('1 2;3 4')
v=numpy.mat('1;1')
print v
print "###################"
for i in range(10):
    z=A*v
    z=z/numpy.linalg.norm(z)
    print z
    v=z
    print "***************"
```

```
In [8]: run mat.py
[[1]
 [1]]
#####################
[[ 0.3939193 ]
 [ 0.91914503]]
***************
[[ 0.41750017]
 [ 0.90867684]]
***************
[[ 0.41586776]
 [ 0.9094251 ]]
```

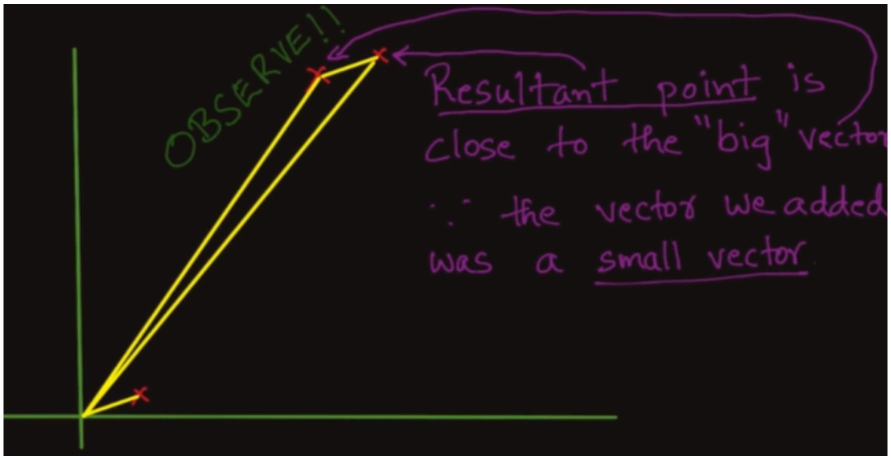
YES IT ALSO CONVERGES!!

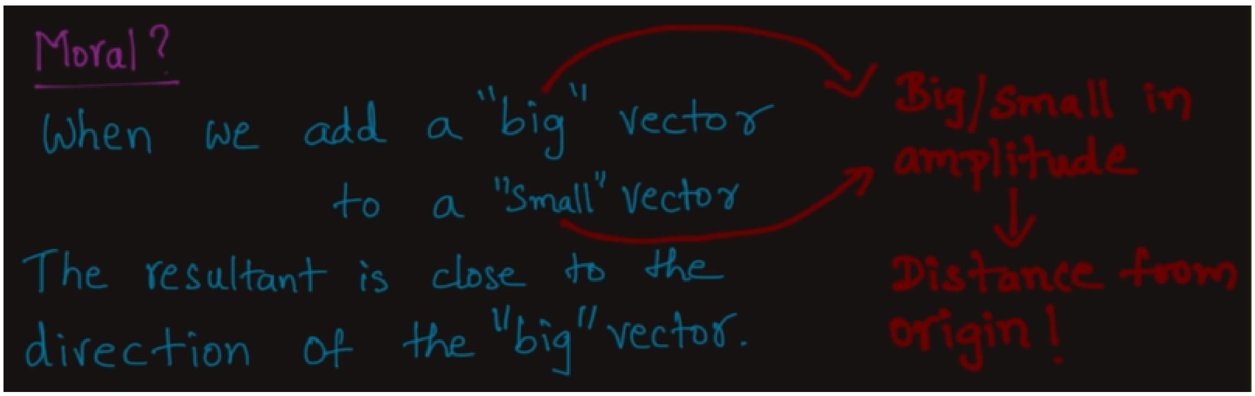
Lecture 111 : Addition of Two Vectors (Pre-requisite 2)

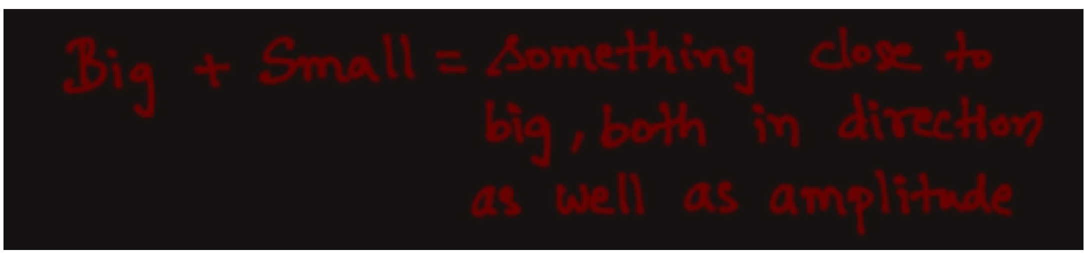




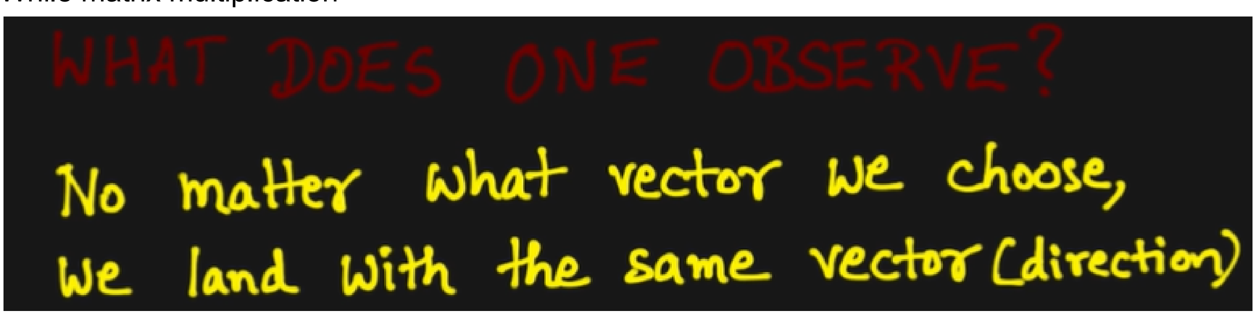
The resultant point is closer to 30,70 than other one

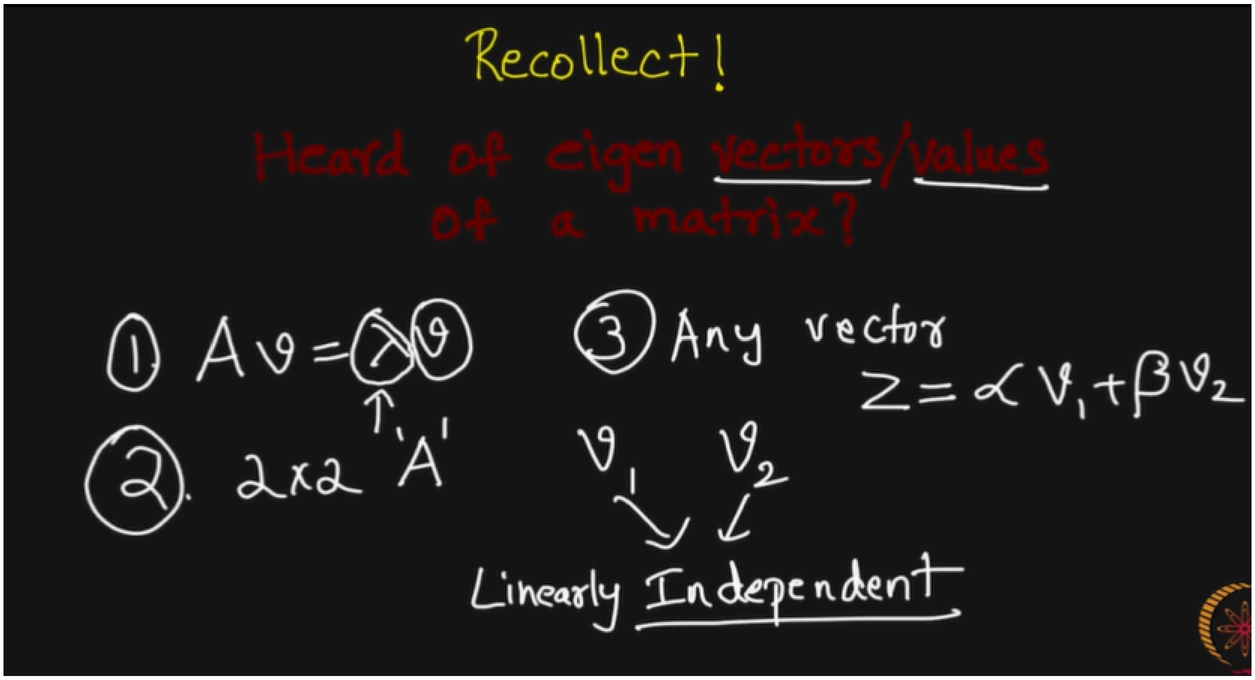


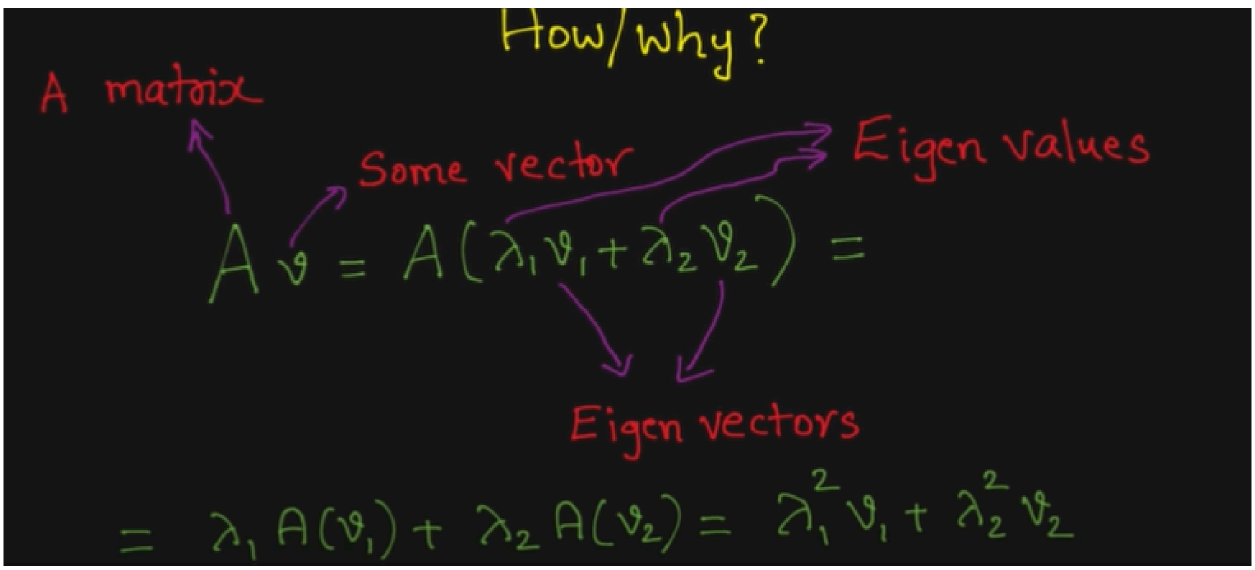




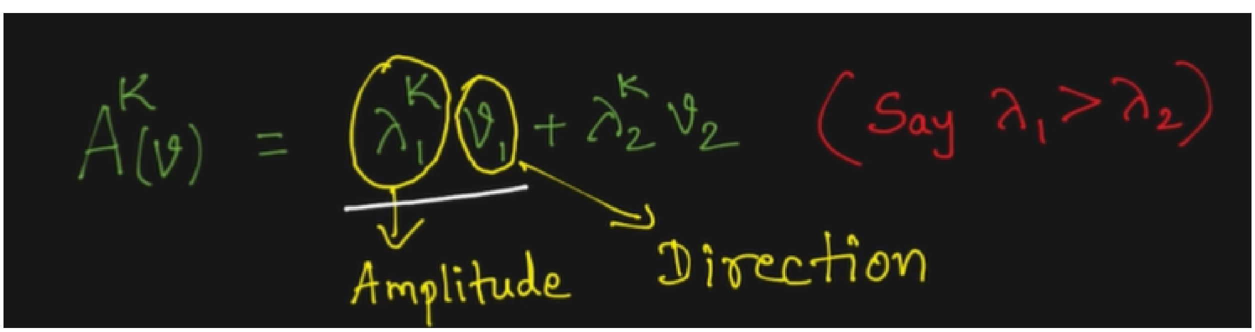
Lecture 112 : Convergence in Repeated Matrix Multiplication- The Details
While matrix multiplication-

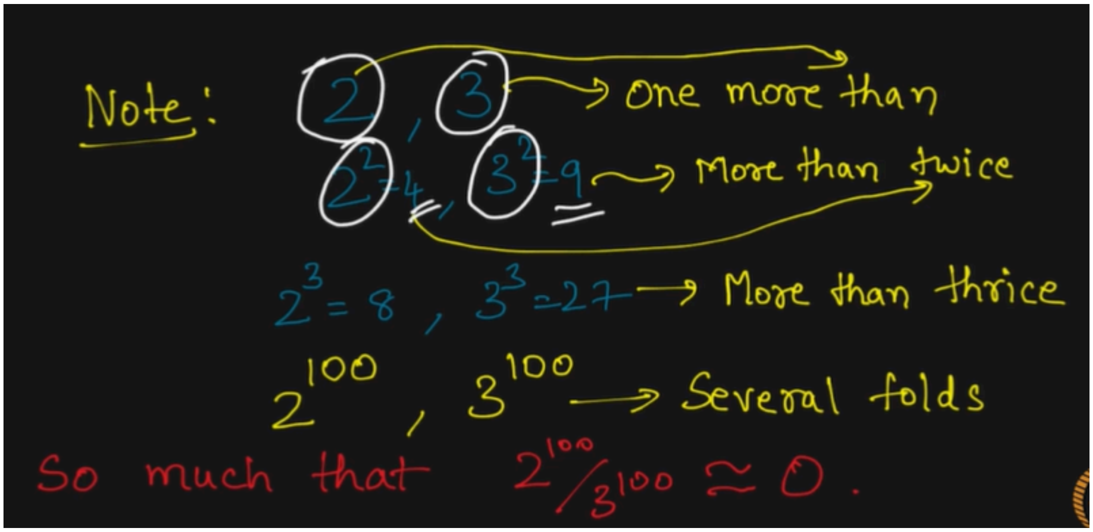


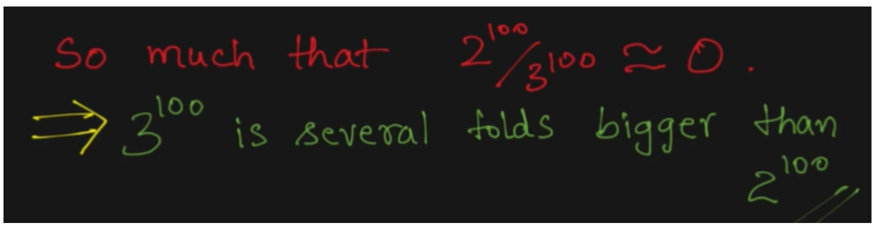




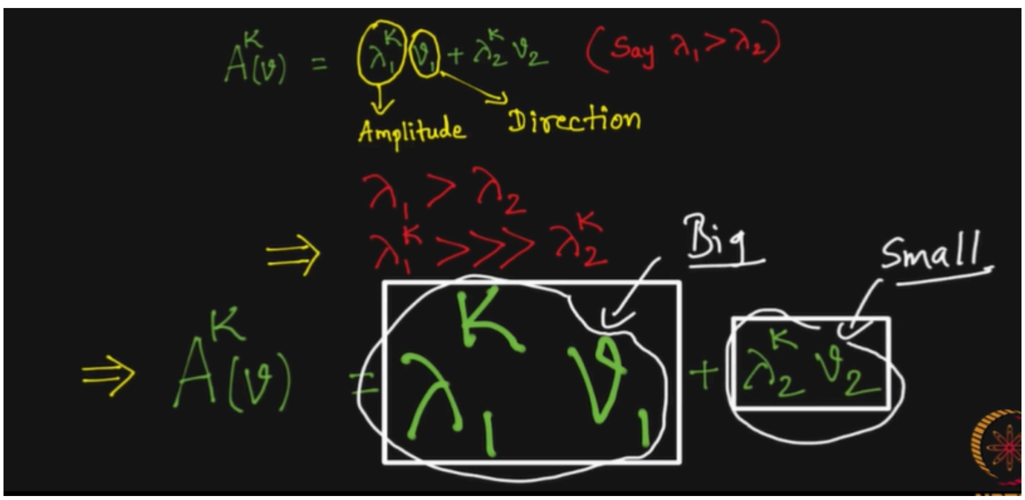
APPLYING A k times on v -



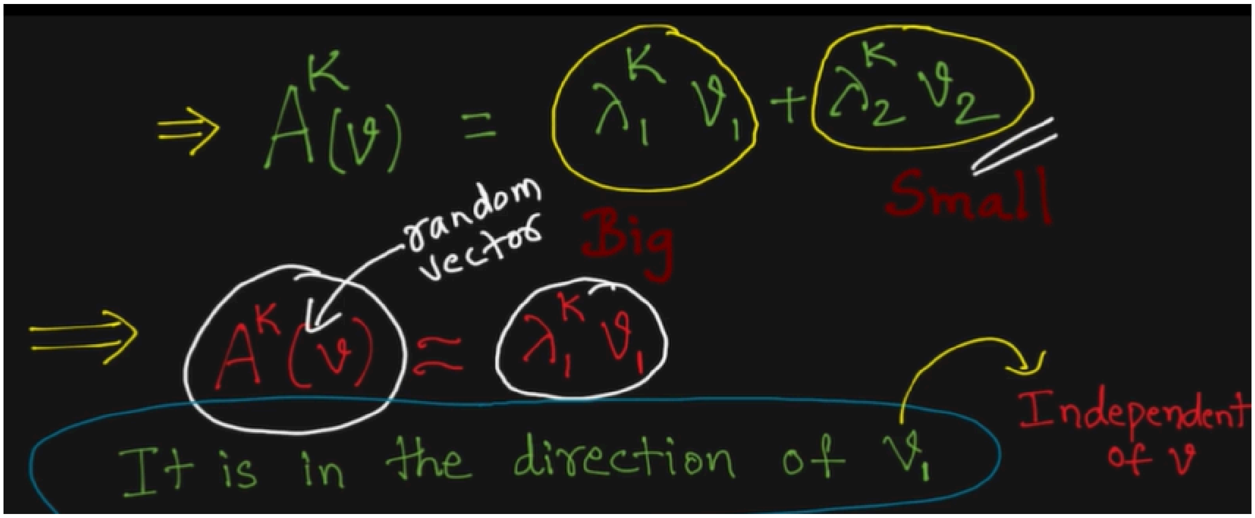




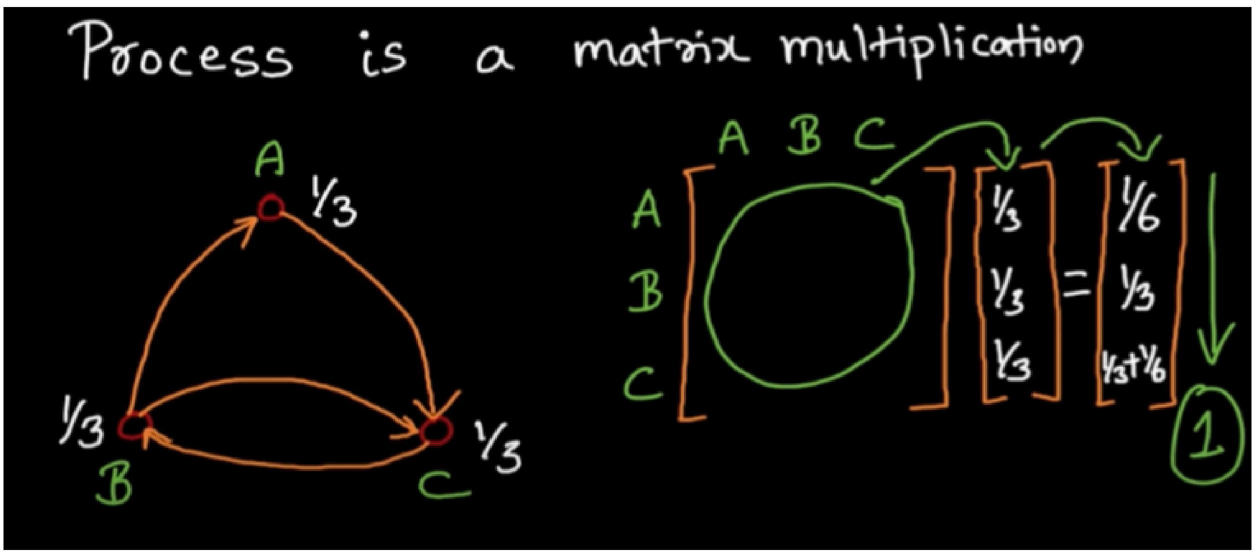
BY the above 2 pics -

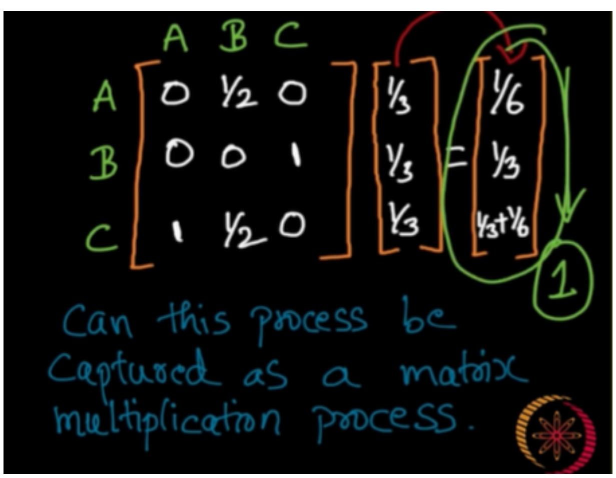


Now here is big vector + small vector

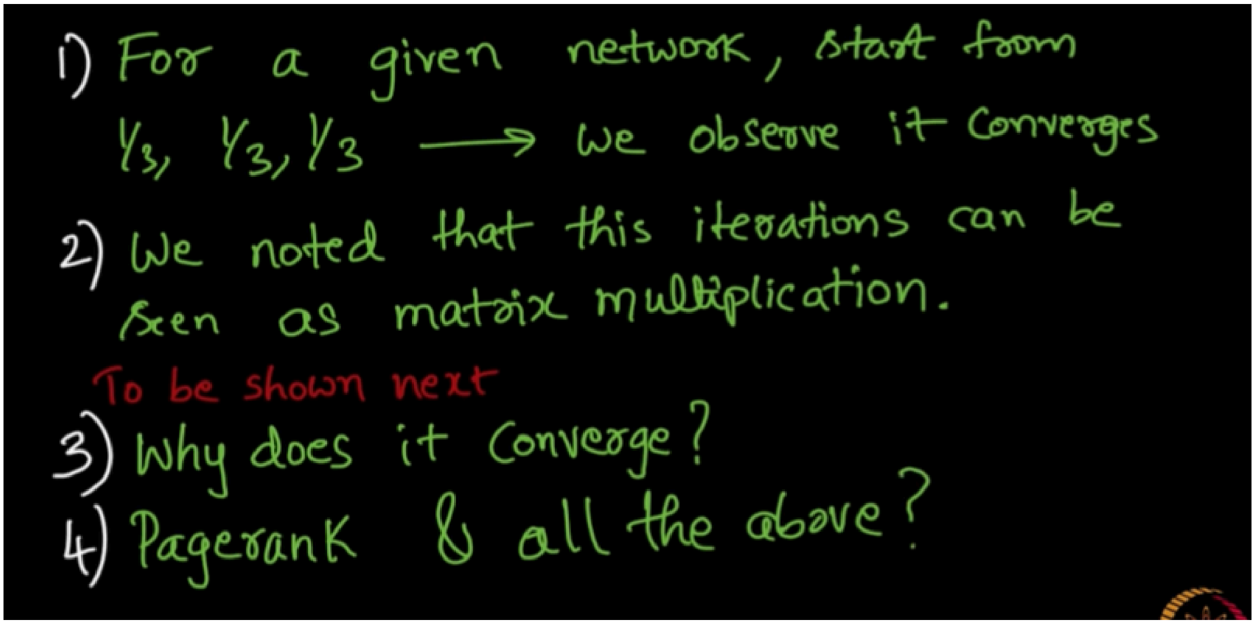


Lecture 113 : PageRank as a Matrix Operation

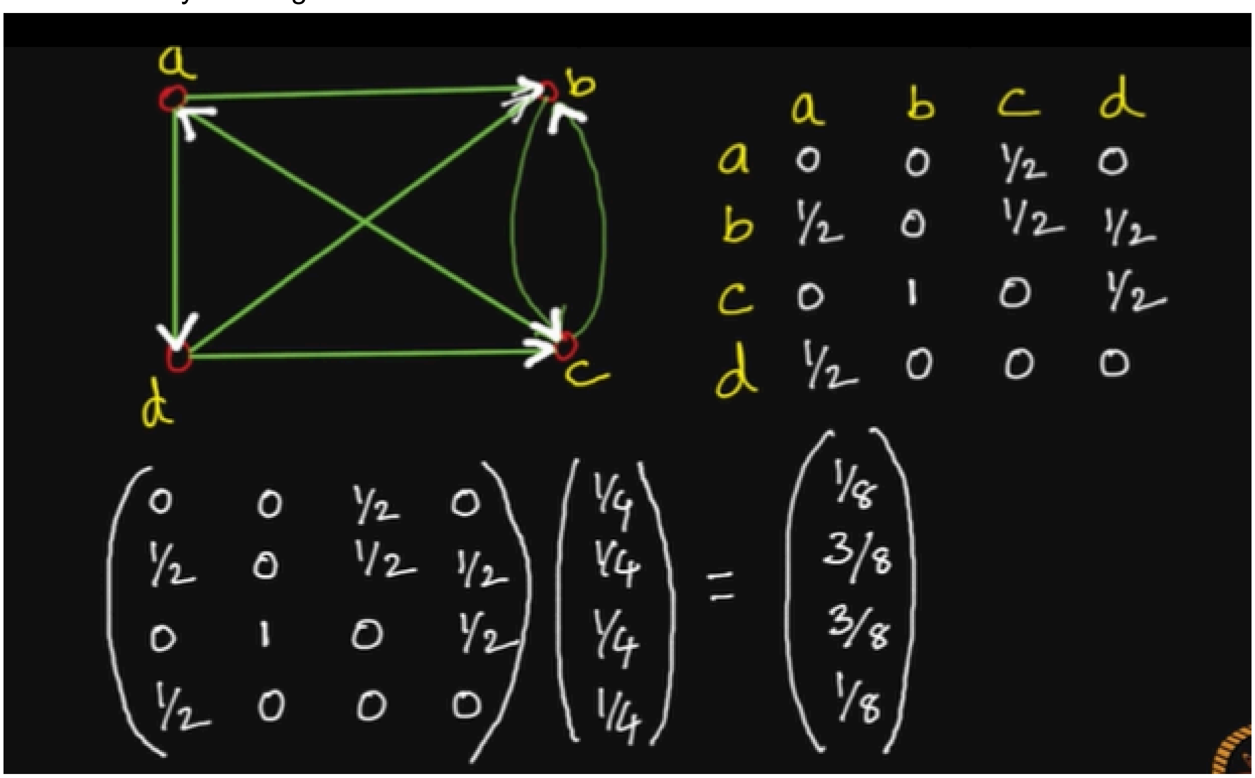




- YES it can be performed by matrix multiplication



Lecture 114 : PageRank Explained
Question - why converge?



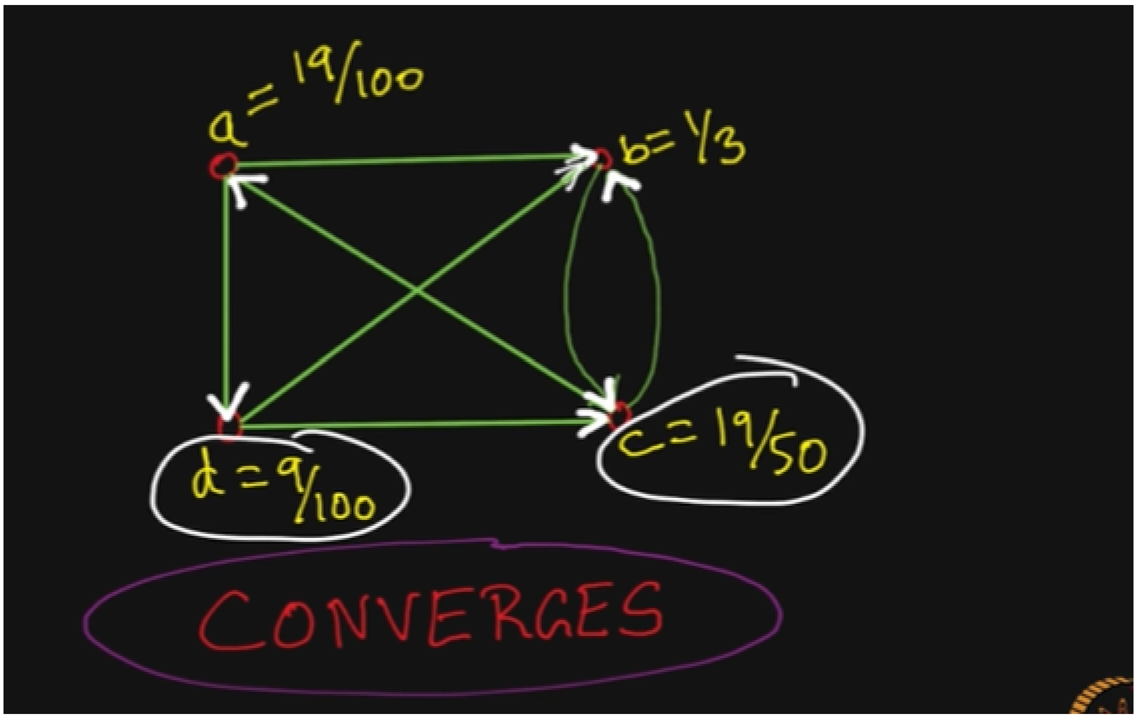
$$\begin{pmatrix} 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 1/2 \\ 0 & 1 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 1/8 \\ 3/8 \\ 3/8 \\ 1/8 \end{pmatrix}$$

Observe

$A\alpha = \alpha$

a	1/4	1/8	.	.	.	.	19/100
b	1/4	3/8	.	.	.	.	1/3
c	1/4	3/8	.	.	.	.	19/50
d	1/4	1/8	.	.	.	.	9/100

Same Values



Why is it converging?

Note

Such a matrix is called a Markov matrix.

$$\begin{pmatrix} 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 1/2 \\ 0 & 1 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 \end{pmatrix}$$

Every column sums to 1

Has a very interesting property

Highest Eigen value is 1

$$\begin{pmatrix} 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 1/2 \\ 0 & 1 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 \end{pmatrix}$$

Has a very interesting property

Highest Eigen value is 1

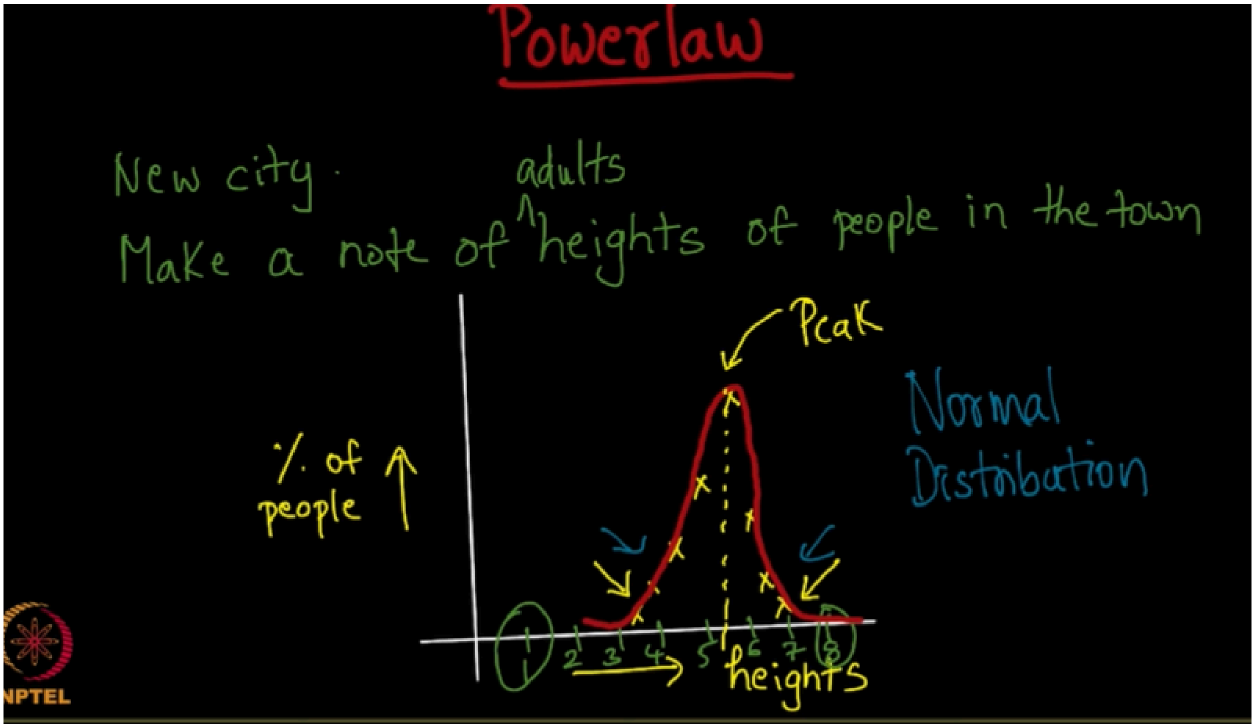
$$\Rightarrow A^K v = \underbrace{\lambda_1^K}_{=1} v_1 + \lambda_2^K v_2 + \dots + \lambda_n^K v_n$$

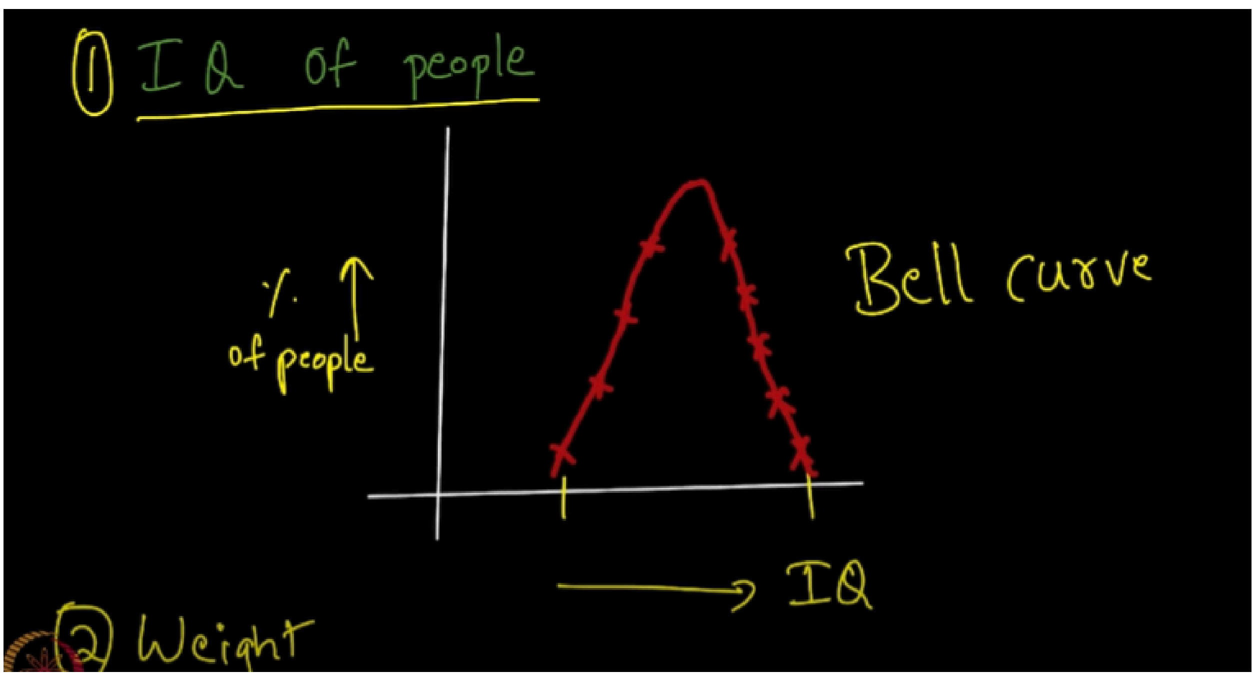
Less than 1

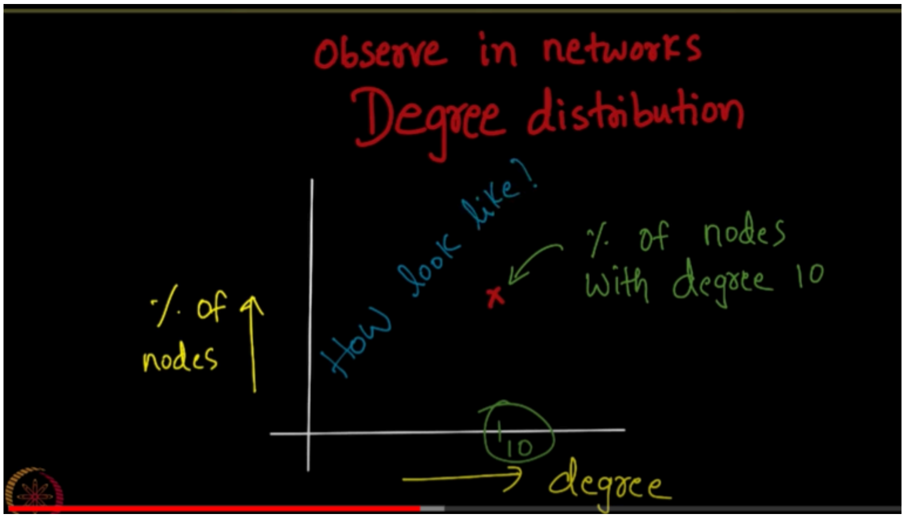
$$\Rightarrow A^K v = v_1$$ ⇝ Eigen vector corresponding to 1

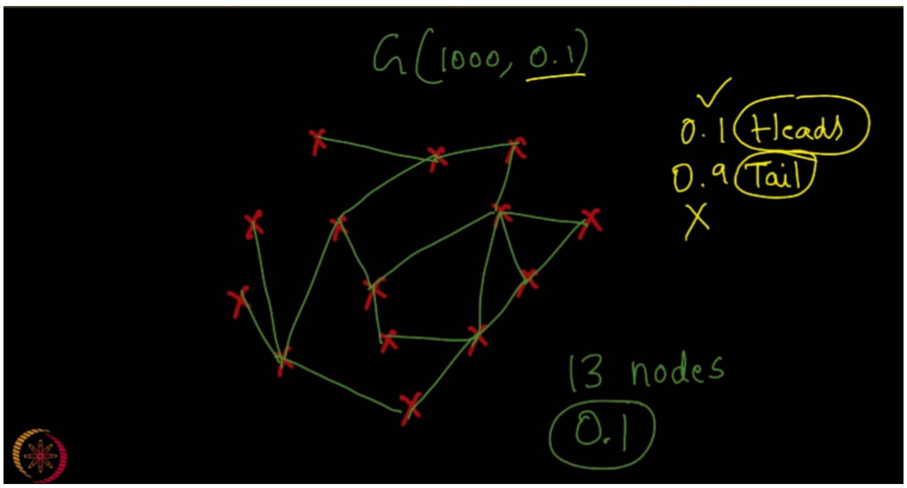
WEEK 9

Lecture 115 - Introduction to Power Law

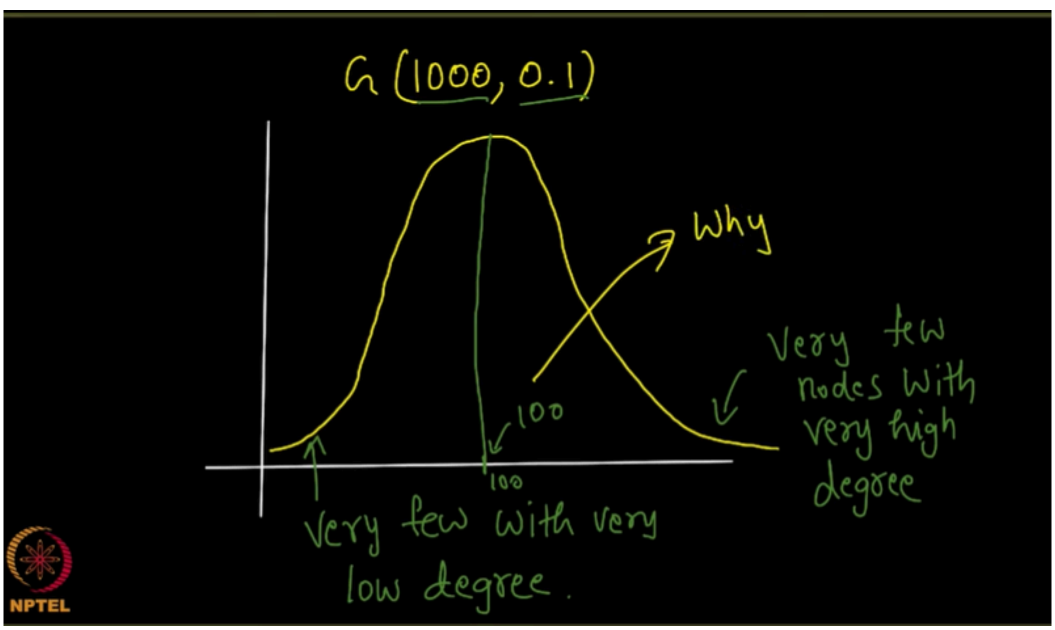




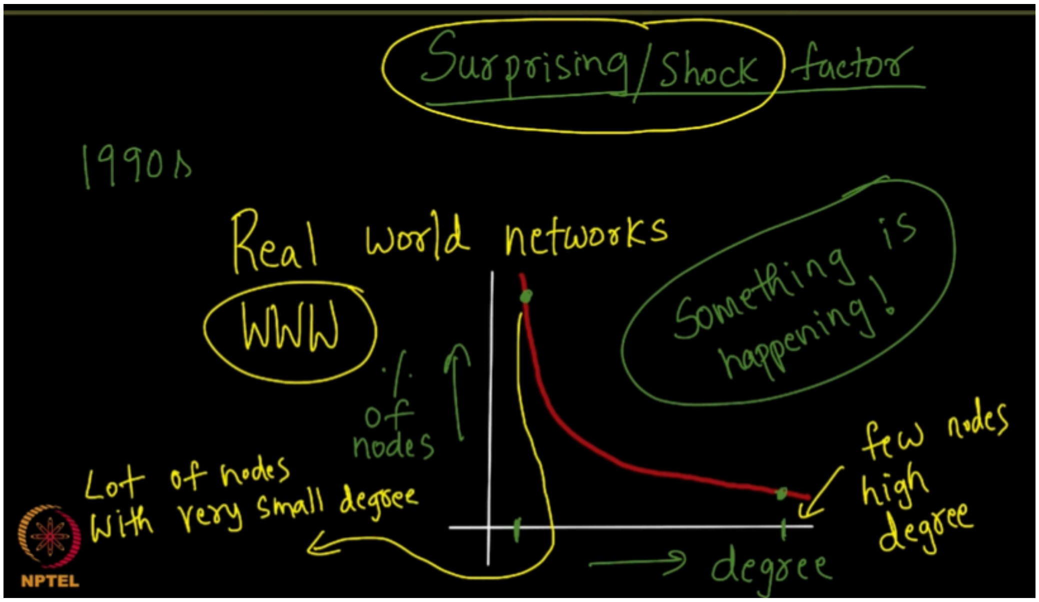




0.1 = PROBABILITY WHETHER TO PUT AN EDGE

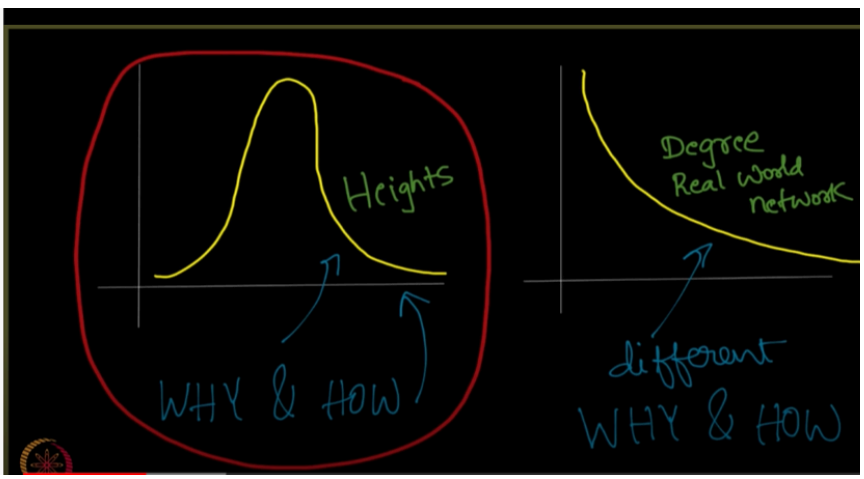


Bell curve again - nothing surprising here

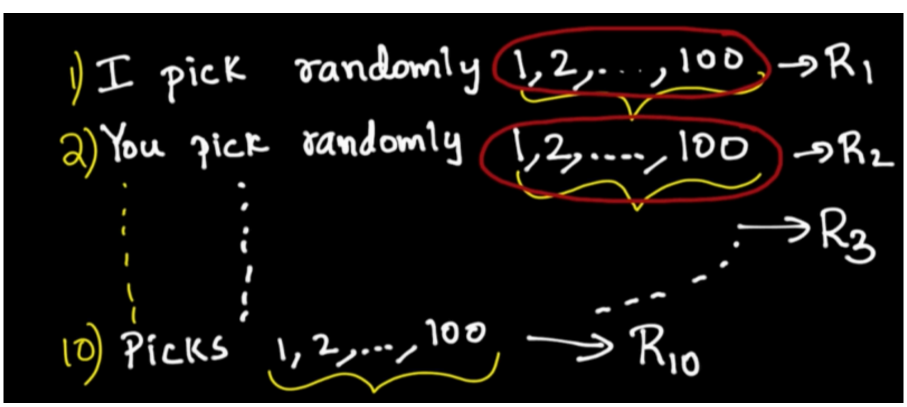


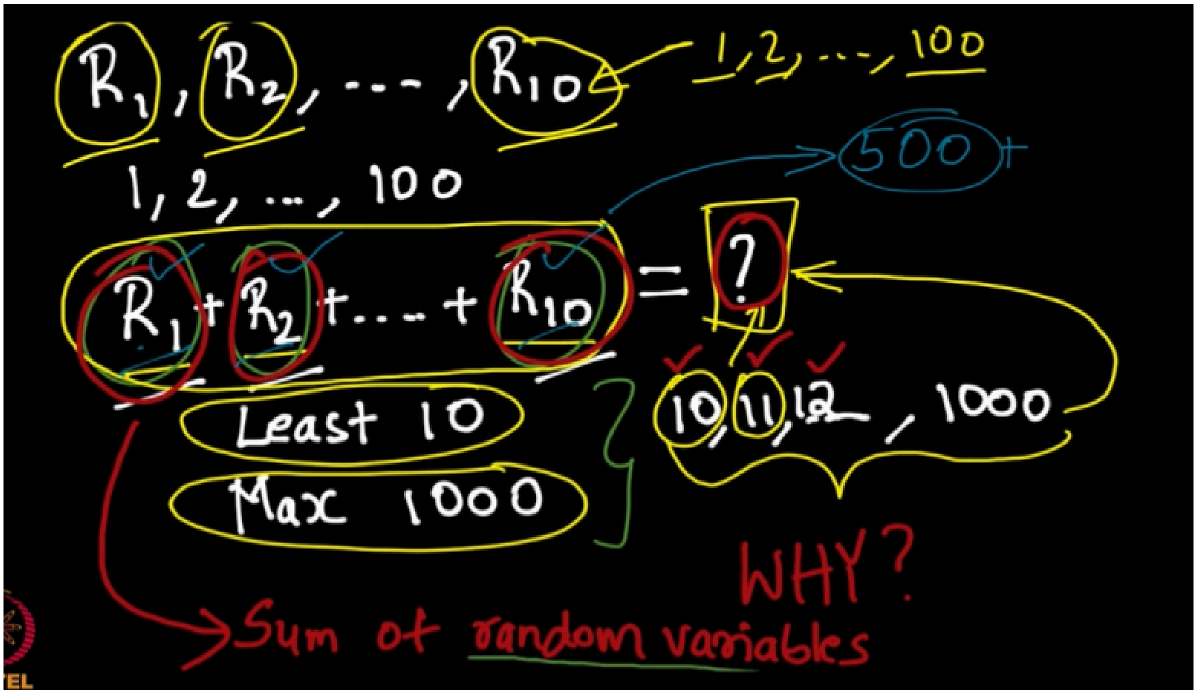
Not a bell shaped curve that was expected .

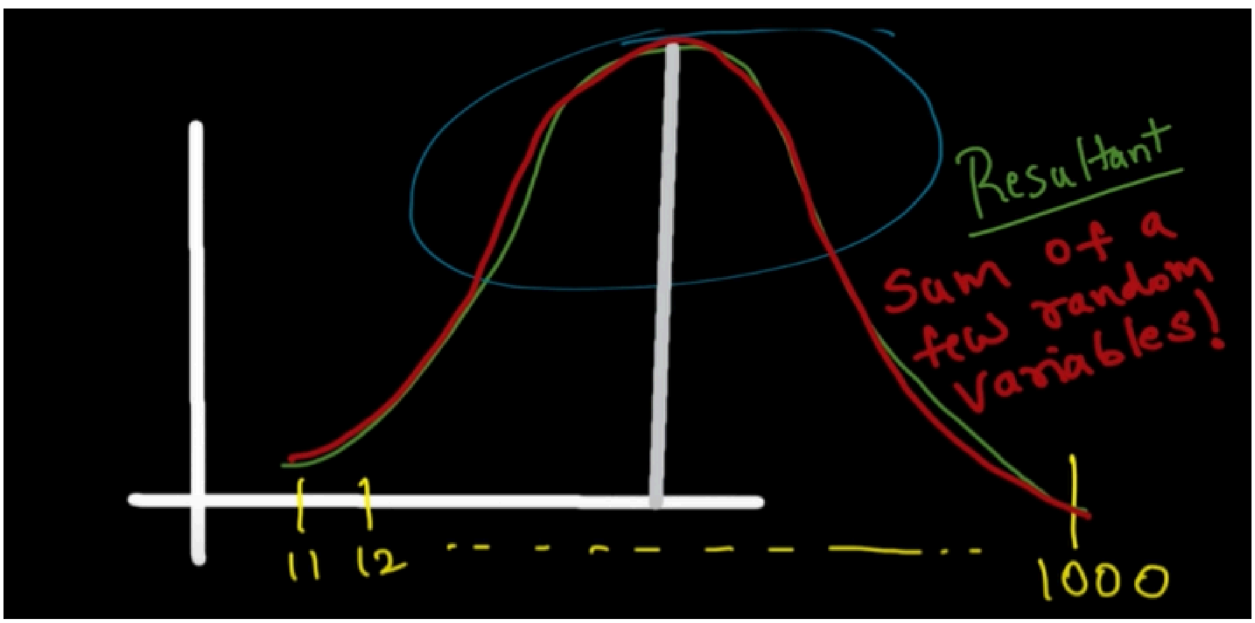
Lecture 116 - Why do Normal Distributions Appear?



Assume -

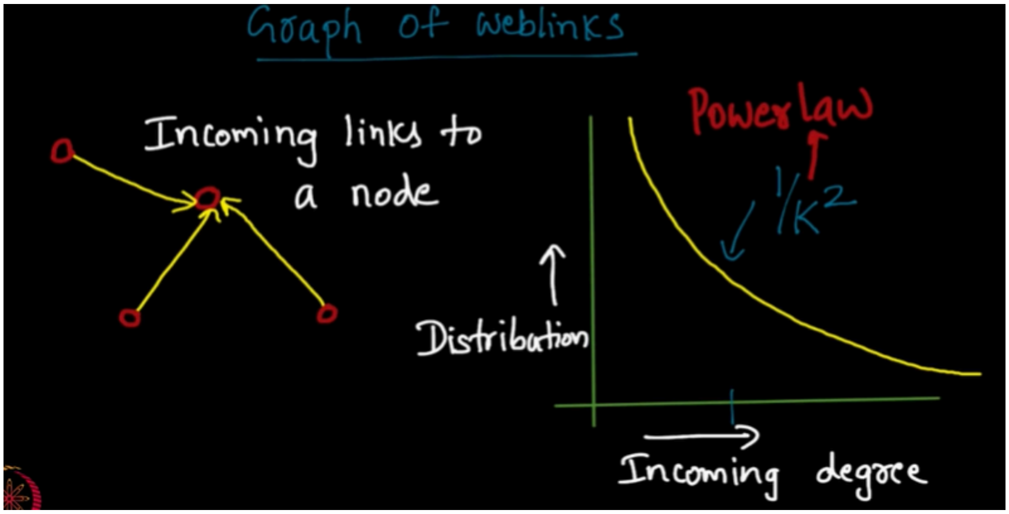


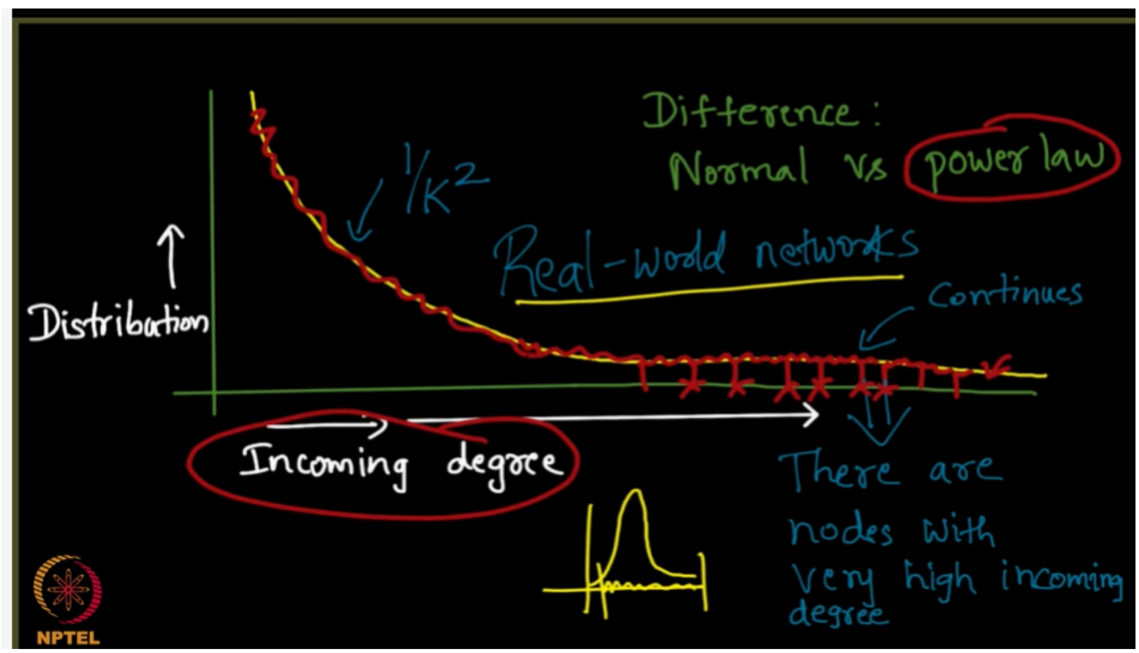




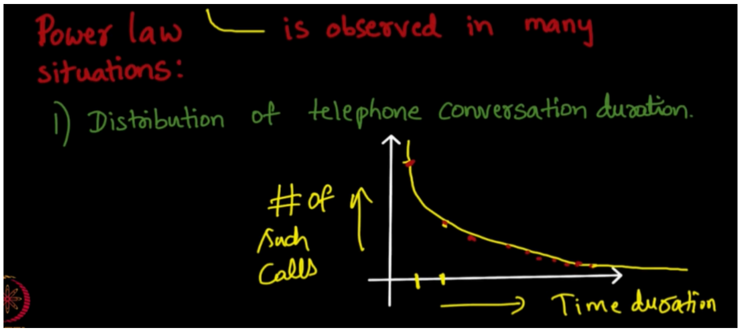
Whenever the resultant is sum of a few random variables then it has a normal distribution - bell curve - **CENTRAL LIMIT THEOREM**

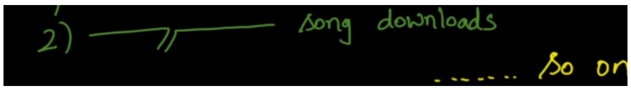
Lecture 117 - Power Law emerges in WWW graphS



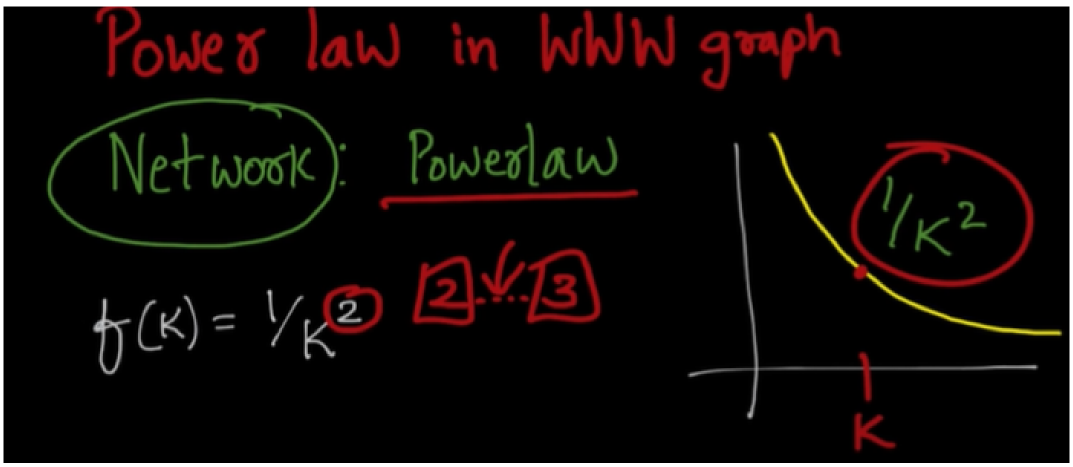


There are many nodes with so high degree were not there in normal curve

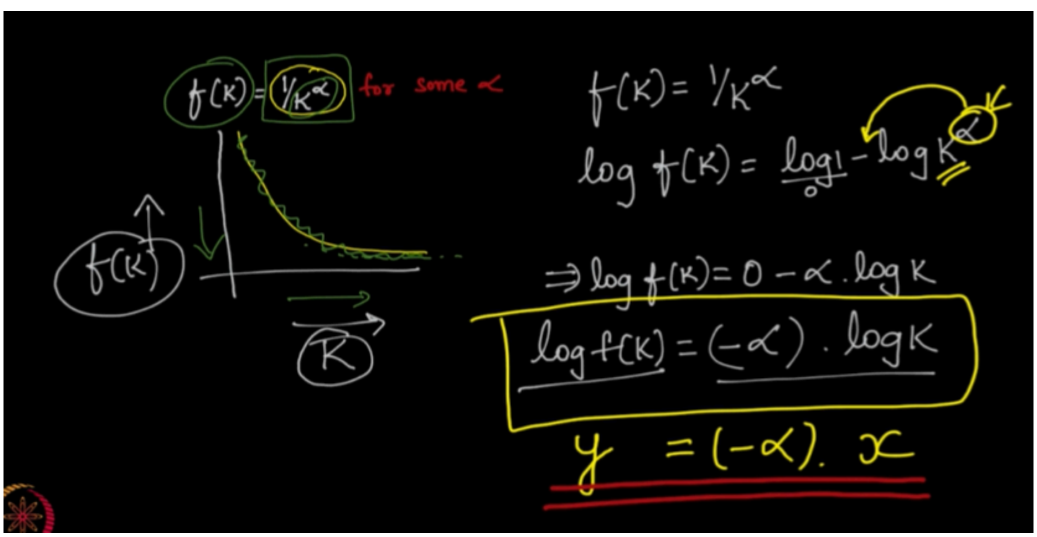


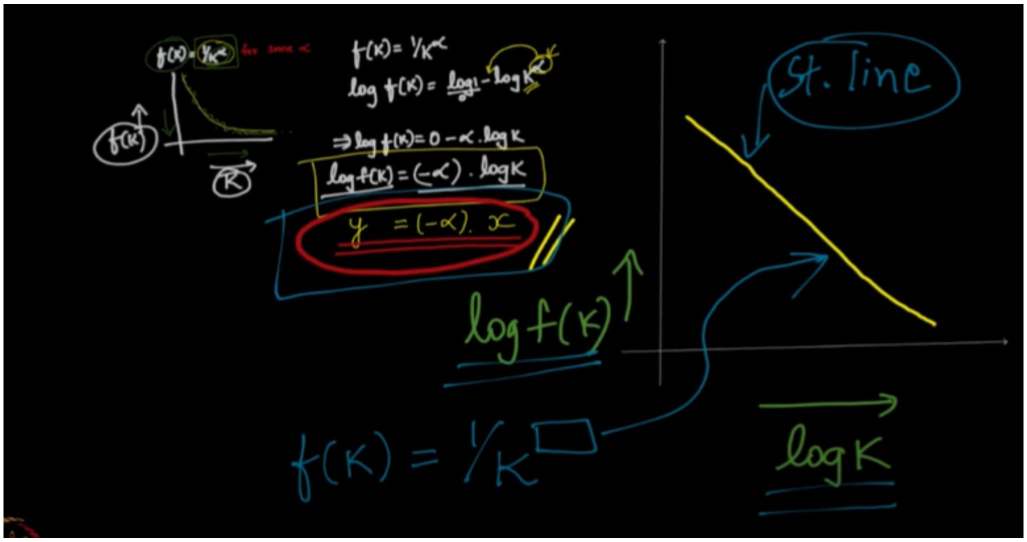


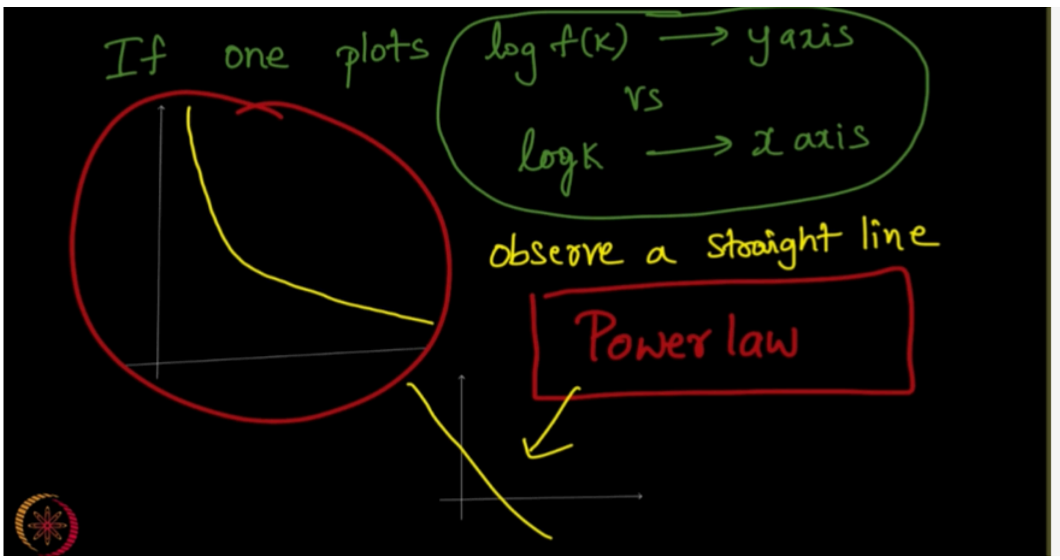
Lecture 118 - Detecting the Presence of Power Law



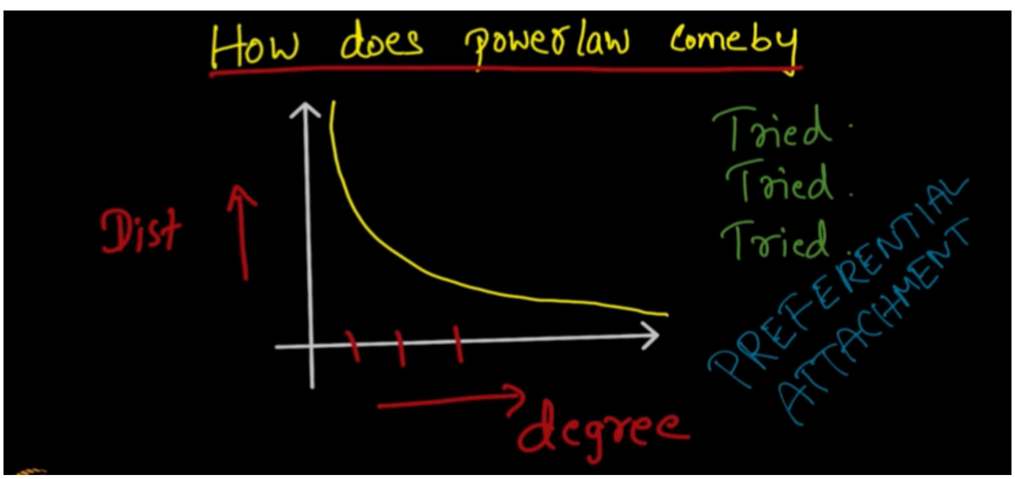
How do we identify whether network exhibits power law. Www graph shows 1/k2. It can be different for other other ntwrks



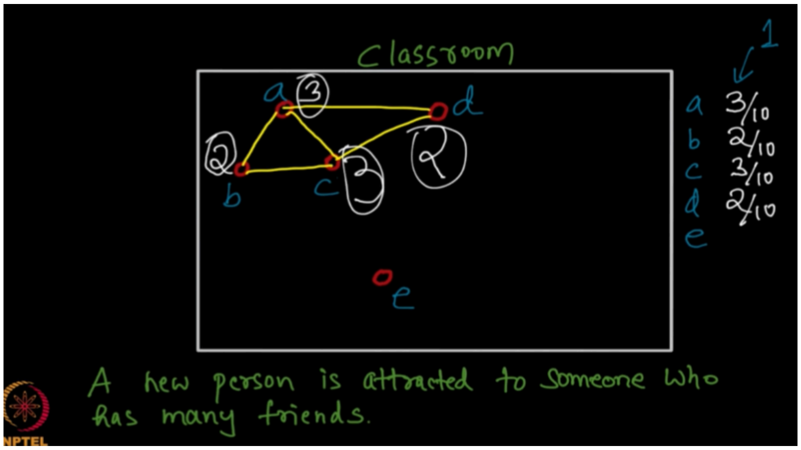




Lecture 119 - Rich Get Richer Phenomenon

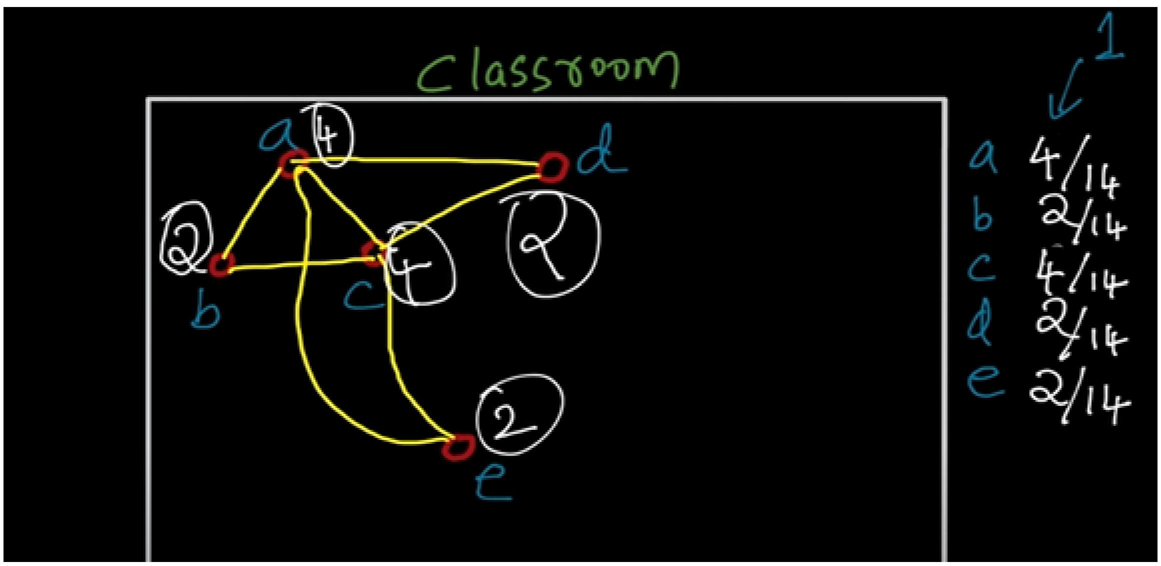


Scientists after trying alot said that powerlaw is happening due to pref attachment hypothesis

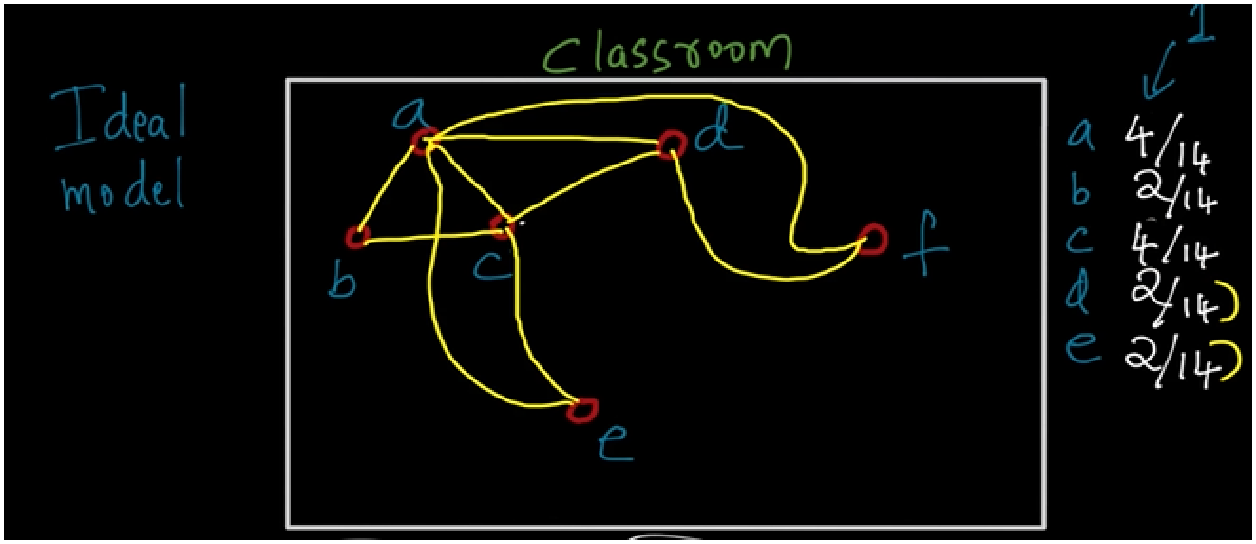


e - new prsn will make friends acc to the given probs - prob is dependent on their already existing degrees

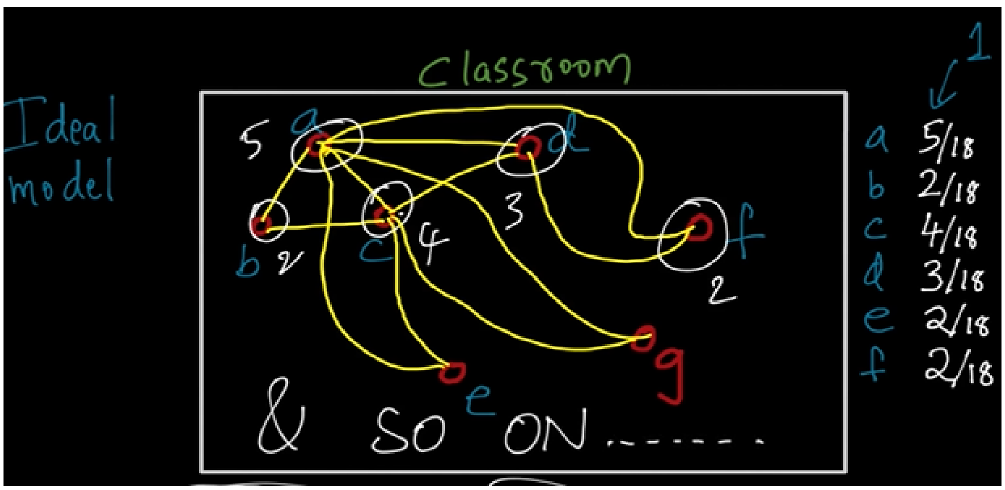
E ends up making friends with C and A . update the degrees

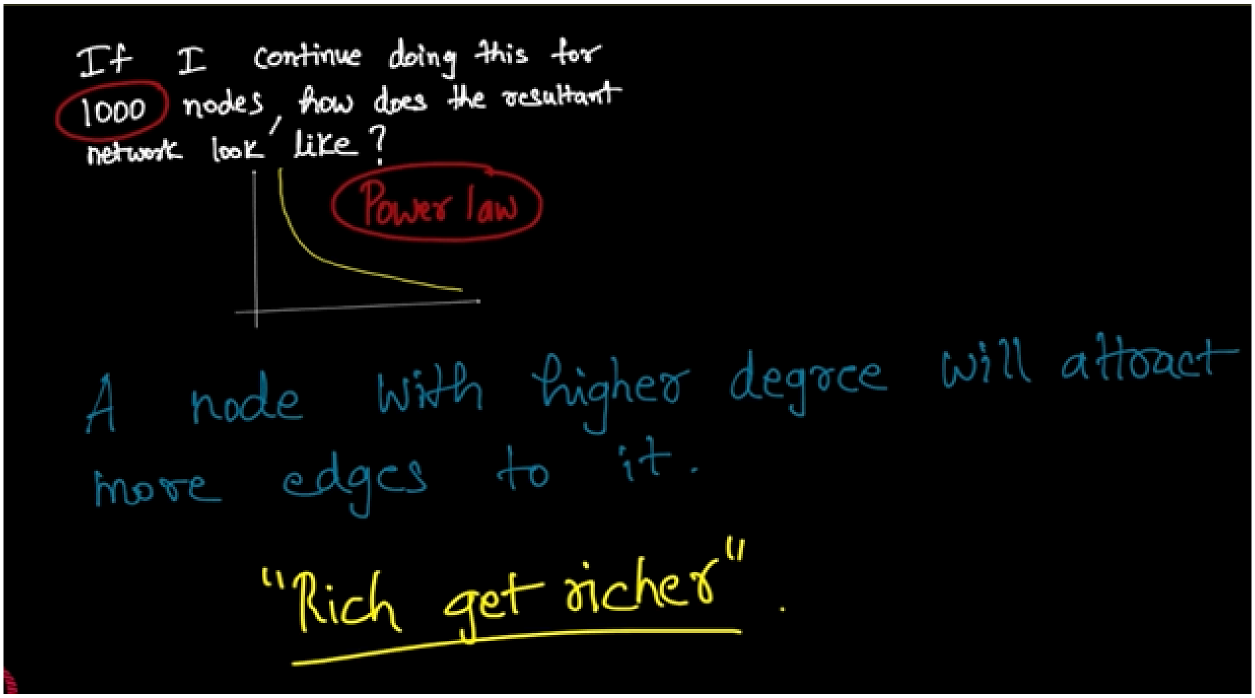


Now a new person F comes - A nd C have high prob but it can also become friend with others. (like d)



Update the degrees. Now new person G comes. It chooses A and C



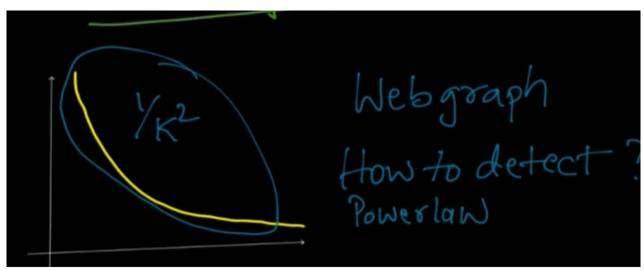


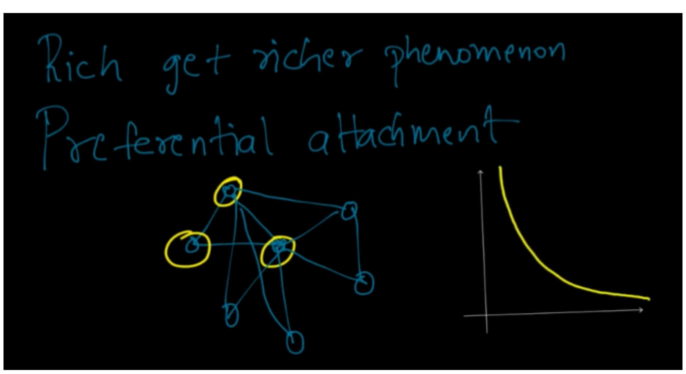
powerlaw. Such a network which grows by respecting the degrees of the existing people and then grow

slowly node by node it follows powerlaw

These rich get richer networks follow powerlaw

Leture 120 - Summary So Far





Lecture 121 - Implementing Rich-getting-richer Phenomenon (Barabasi-Albert Model)-1

It is also called Matthew effect and it is also

called Preferential Attachment. Now, there is a particular model that is called Barabasi Albert

model which is precisely based on this method that is called preferential attachment . So,

Main idea -

1. We start with m_0 nodes, the links between which are chosen arbitrarily, as long as each node has at least one link.
2. The network develops following two steps:
   1. **Growth:** At each timestep we add a new node with m ($\leq m_0$) links that connect the new node to m nodes already in the network.
   2. **Preferential attachment:** The probability that a link of the new node connects to node i depends on the degree of node i.

Steps for Implementation

- Take n, i.e. total number of nodes, from the user.
- Either take m, i.e. the number of edges to be connected to the new node, from the user, or decide it yourself based on n. In our implementation we will decide it ourselves as follows:
  - (m0 is the initial number of nodes that should have alteast one link. m should be less than or equal to m0).
  - We will take m0 to be any random number between 2 to n/5. (You can use any measure.)
  - We will take m to be one less than m0.
- Add the rest n-m0 nodes. Add edges to these n-m0 nodes based on 'preferential attachment'.

For all $n - m_0$ nodes, repeat the following:

- Add the node.
- To add the edges, do the following:
  - Preprocessing:
    - Get a dictionary of degrees (since preferential attachment will happen based on degrees).
    - Maintain a dictionary of probabilities. (The probabilities have to be assigned based on degrees. More the degree, more the probability. Precisely, probability[i] = Degree[i]/Sum(degrees of all the nodes)).
    - Maintain a list of lists for maintaining cumulative node probabilities.(This is for choosing a node based on probabilities)
      For Example,
      Probabilities = [0.2, 0.3, 0.5]
      $-->$ Cumulative Probabilities = [0.2, 0.5, 1.0].
  - While *edges_added* are not equal to m:
    - Choose a random number from 0 to 1.
    - Whichever node has cumulative probability more than this random number, the edge will be connected to that node, if not already added.
      If r = 0.4
      $-->$ Node 2 will be chose out nodes 1, 2 and 3.

Lecture 122- Coding

Lecture 123 - Implementing a Random Graph (Erdos- Renyi Model)-1

1. Take n, i.e. total number of nodes, from the user.
2. Take p, i.e. the value of probability from the user.
3. Create an empty graph. Add n nodes to it.
4. Add edges to the graph *randomly*.

For random adding of edges

1. Take a pair of nodes.
2. Get a random number r.
3. If r is less than p: Add this edge, else ignore.
4. Repeat step 1 for all possible pair of nodes.

Degree distribution - normal distribution after coding

Lecture 125 - Forced Versus Random Removal of Nodes (Attack Survivability)

In this video we are going to implement Forced
Versus Random Failure of Nodes which is also

called attack survivability. We are going to
take a network and we will be removingnodes

from this network and we willobserve the effect
effect of removal of nodes from the network. We

will be using two strategies to remove the
nodes. In the first strategy we will remove

the nodes randomly and in the second strategy we
willremove the nodes which are highly connected.

We will observe that when we remove the nodes
which are highly connected, it requires very less

iterations for thegraph to become disconnected.
This sort of information can be used by some

agent for example, if he has the information of
who is more connected and who is not,he if in

case he wants to damage the network for example,
by spreading some sort of virus, he will not be

network becomes infected to a greater extent.
So, there is one thing we will observe and

implement and we will also see one more thing in the context of the kinds of graphs. For example,

the first thing that we checked on real world networks,real world networks sort

of followpreferential attachment where there are there is existence of hubs in

the network which are responsible for connection of most part of the network. So, when you remove

then thethe graph becomes, becomes disconnected. We are going to observethe same thing that is the

removal of nodes on random networks as well and we will observe that it does not make much

difference on a random network when we remove the nodes randomlyas compared to when we remove

the nodes selectively; that is weremove the nodes which are having high degree.

That is because there is there are no hubs in random networks which are responsible for the

connection ofmost part of the network. So,the number of iterations that it will require in

thefirst case that is random removal and the second case that is selective

removal. In the case of random networks will not make much difference.We are going to implement that and we will observeum the number iterations that are required. Let us

After implementation -

FOR REAL WORLD NETWORKS

So, in this case you seerandomly removal took 16 iterations and the selective removal took only 1.

```
Is G1 connected? True
Is G1 connected? True
Is G1 connected? False
nodes removed when randomly chosen 13

###Selective removal###
Is G2 connected? True
Is G2 connected? False
nodes removed when selectively chosen 1
anamika@anamika-Inspiron-5423:~/NPTEL/WEEK wise/
```

FOR RANDOM NETWORKS

```
nodes removed when randomly chosen 985
```

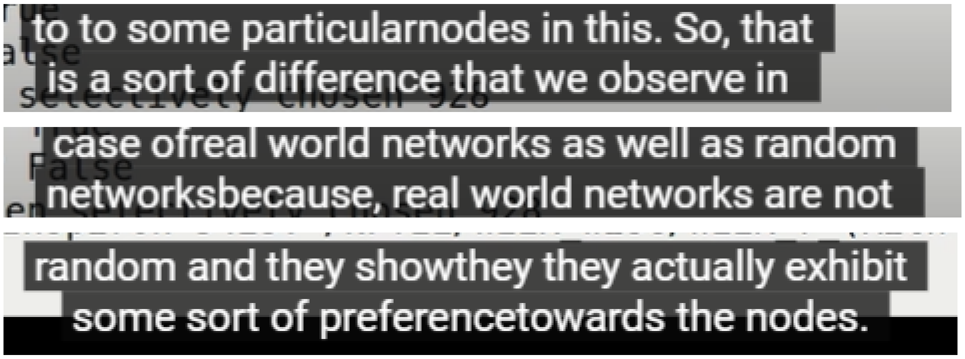
```
nodes removed when selectively chosen 928
```

Nowthat is happening because, there are no hubs inrandom networks which are connectingum a lot of

nodes. ah. So, all the nodes are sort of equally likely because,um equally likely in the sense

ofthe number ofnodes that they are connected to. um. So,there is no preferencebecause,

the edges were added randomly and there is no preference towards addition of nodes



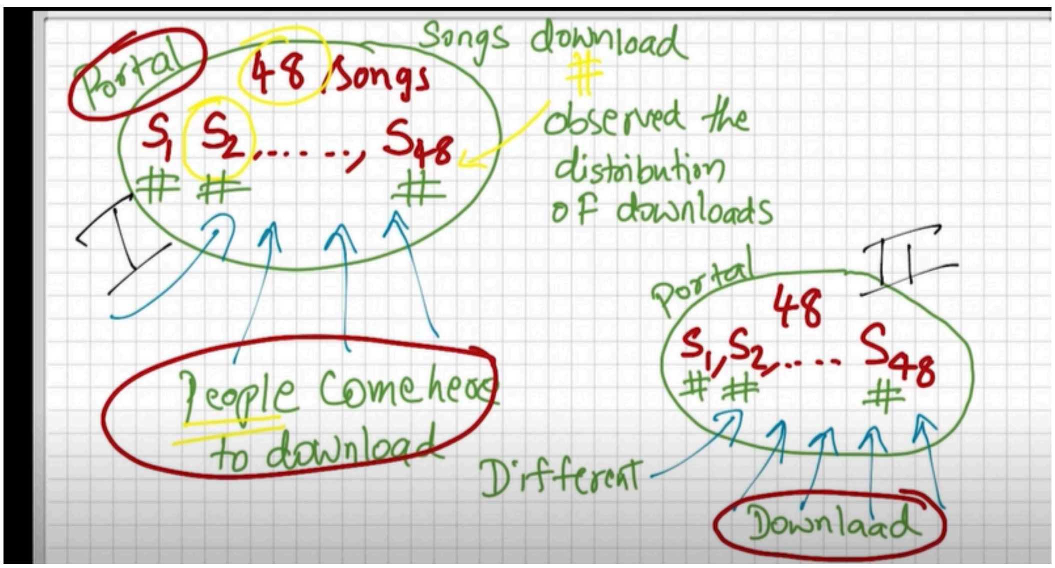
WEEK 10

Lecture 126 - Rich Get Richer - A Possible Reason

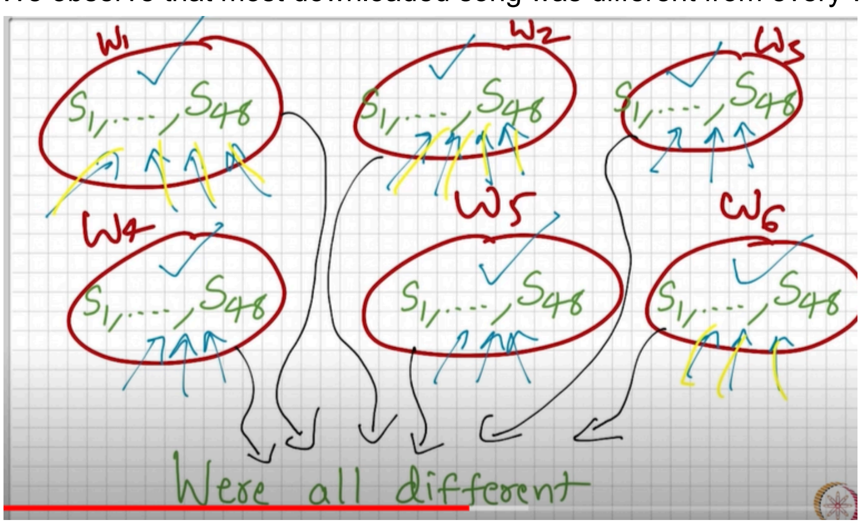
How did the 1st person become rich ? by luck or by hard work?

Perform an experiment -

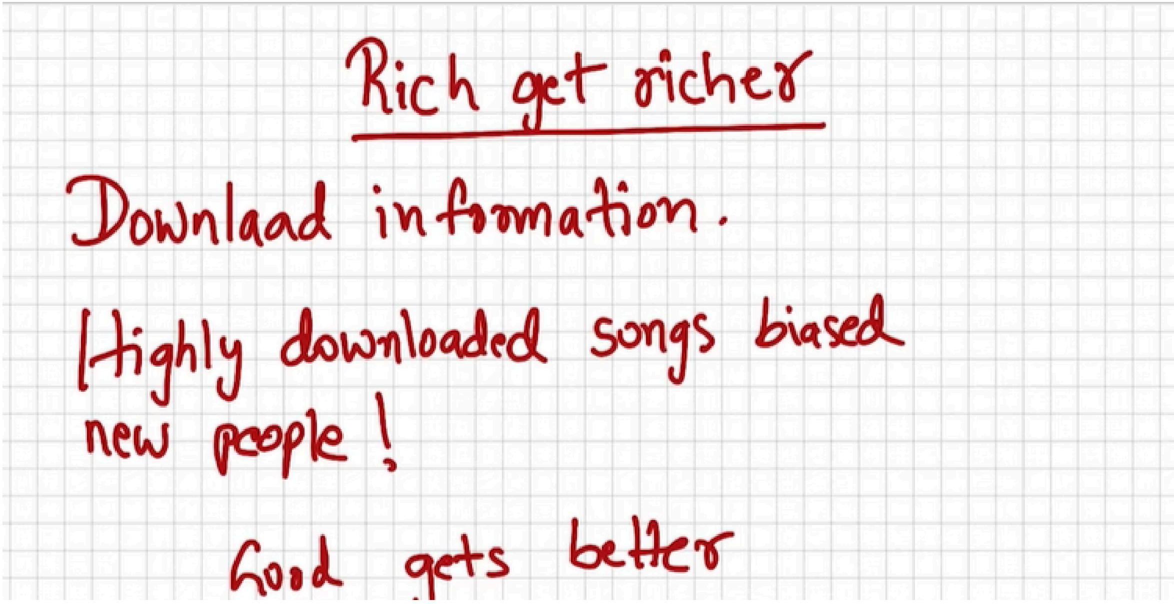
Many different worlds, having 48 songs,different people are coming to download the songs

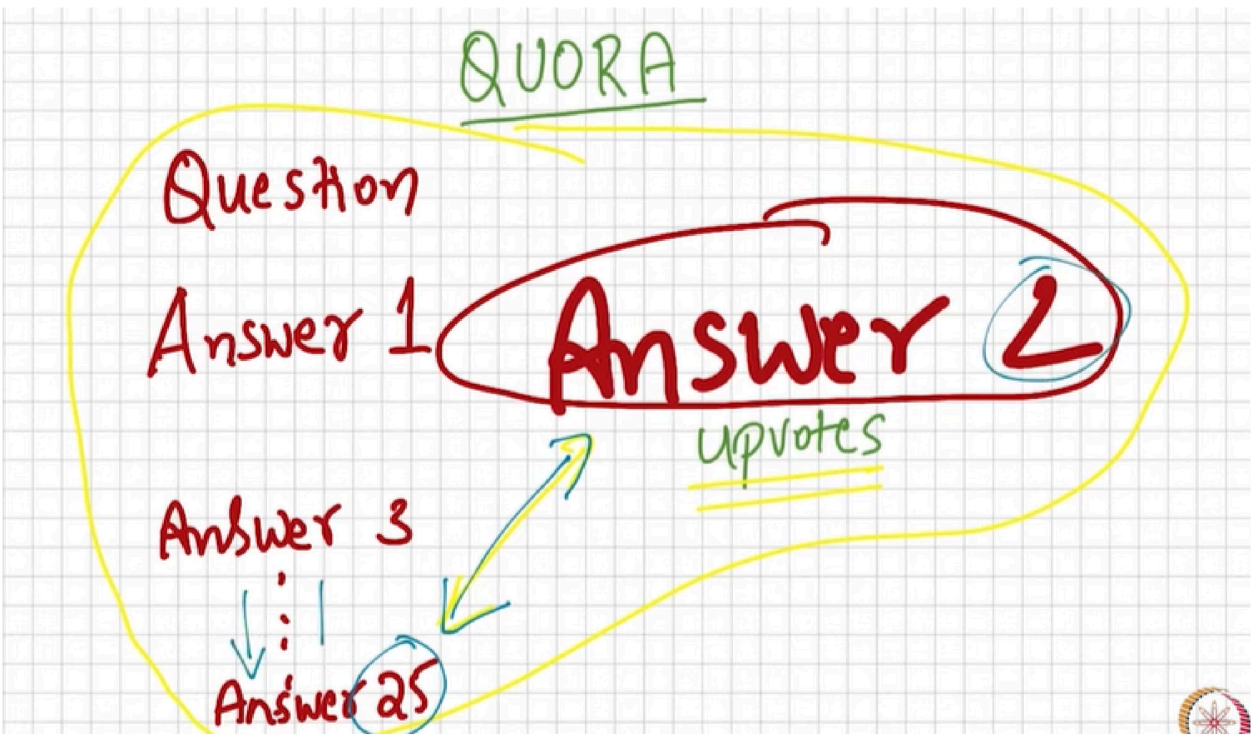


We observe that most downloaded song was different from every world.



Reason ?





When a new ans25 comes (even if it is more apt) , no one is able to see it bcz it is flushed down as the no. of upvotes are less

"Rich"? get richer

Rewind history?

1^{st} ranked people aren't same

EXPERIMENT

It is not imp the same 1st rank is there everytime if we rewind the history

Questions that arise -

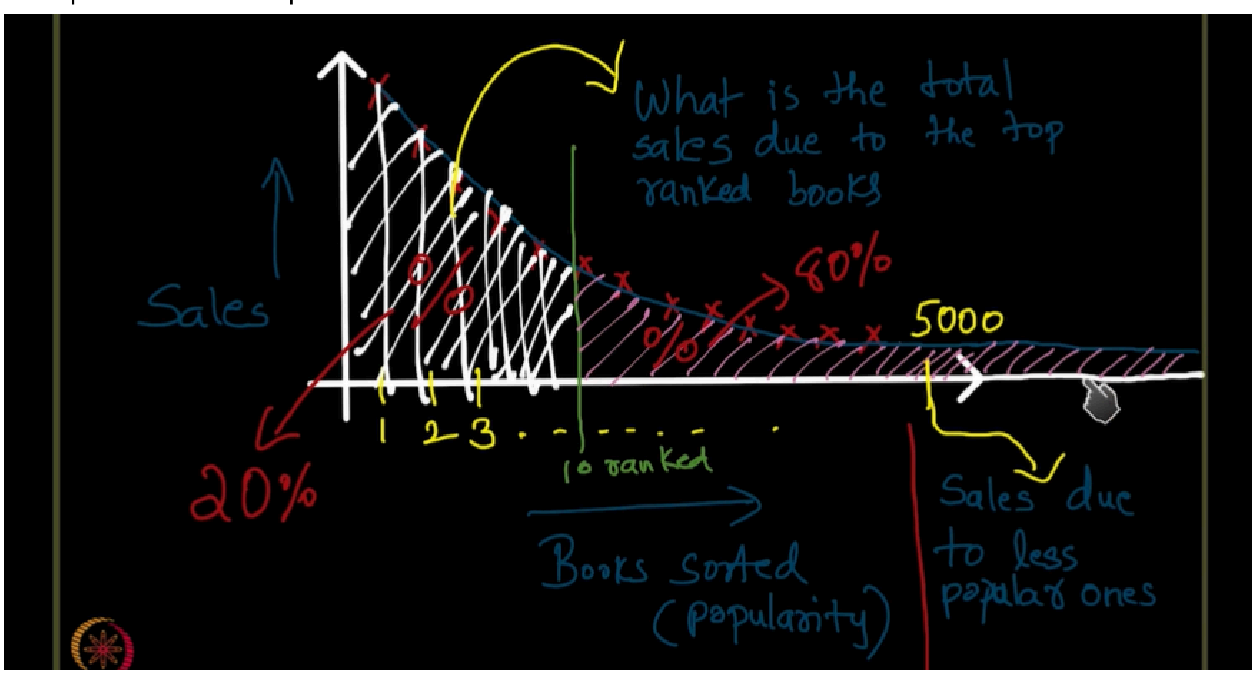
1) How can one detect the right talent by subtracting out the popularity obtained by RGR phenomenon.
2) RGR
How does one judge the true value of an entity

1) RGR the reason for emergence power law.
2) RGR adds noise to popularity!
How to remove.

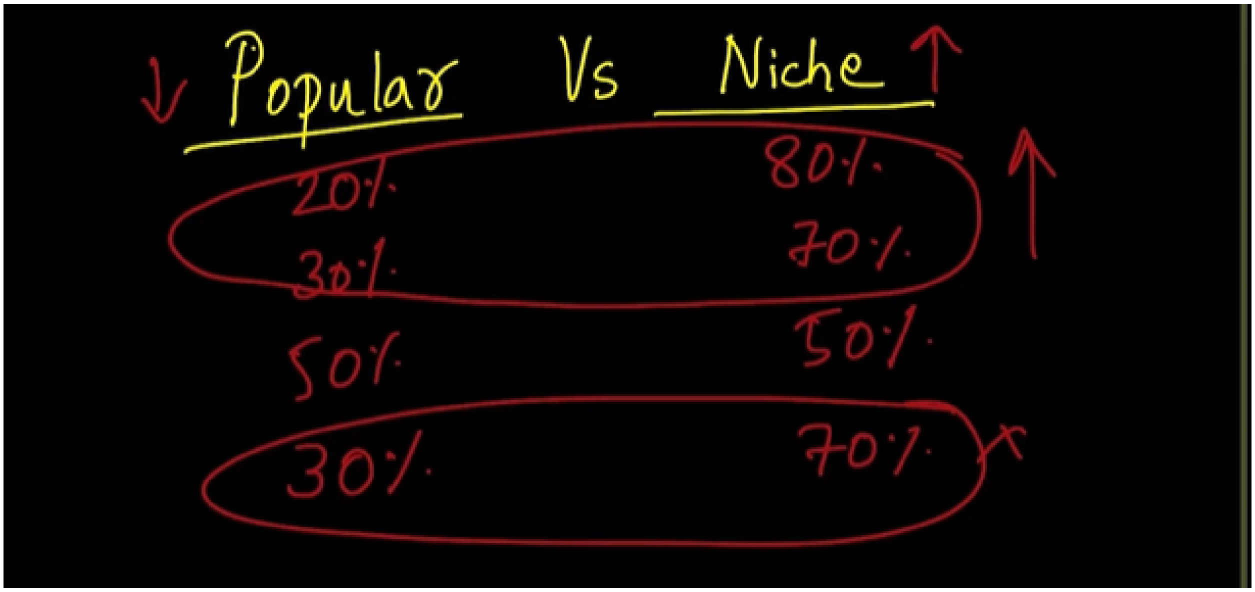
Lecture 127 - Rich Get Richer - The Long Tail

why not just keep the 10 percent of it (the books- the bestsellers)
and discard the remaining 90 percent;it

This is a wrong hypothesis for the business.
Total profit of 1st 10 top books=area below the curve

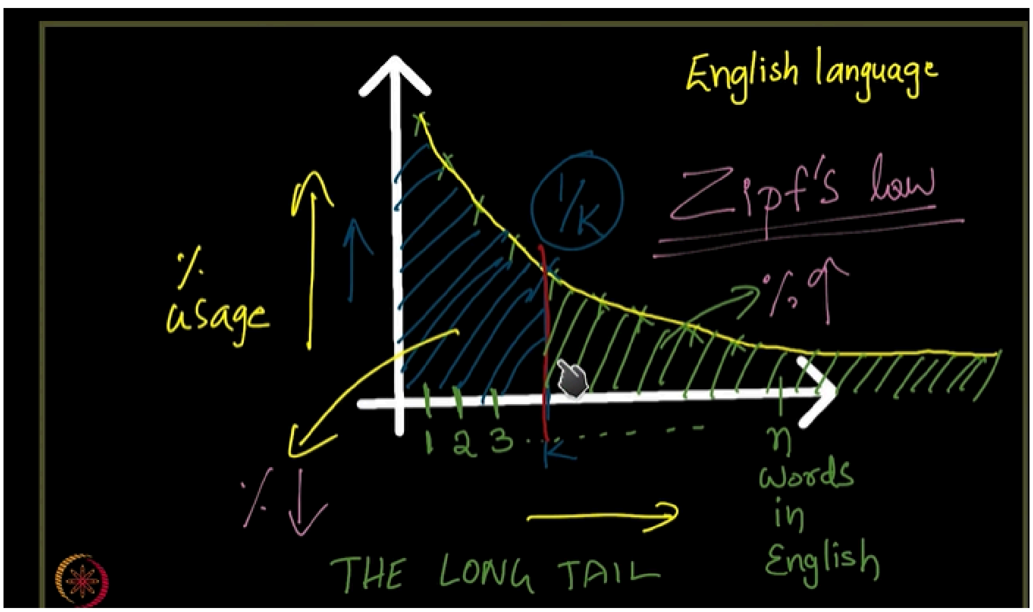


After a prd of time, the popular bestsellers move to classic sellers and come in the 80prcnt region



This kind of thing can also happen in the network nodes . a node may not continue to be the most popular forever

In english language ,



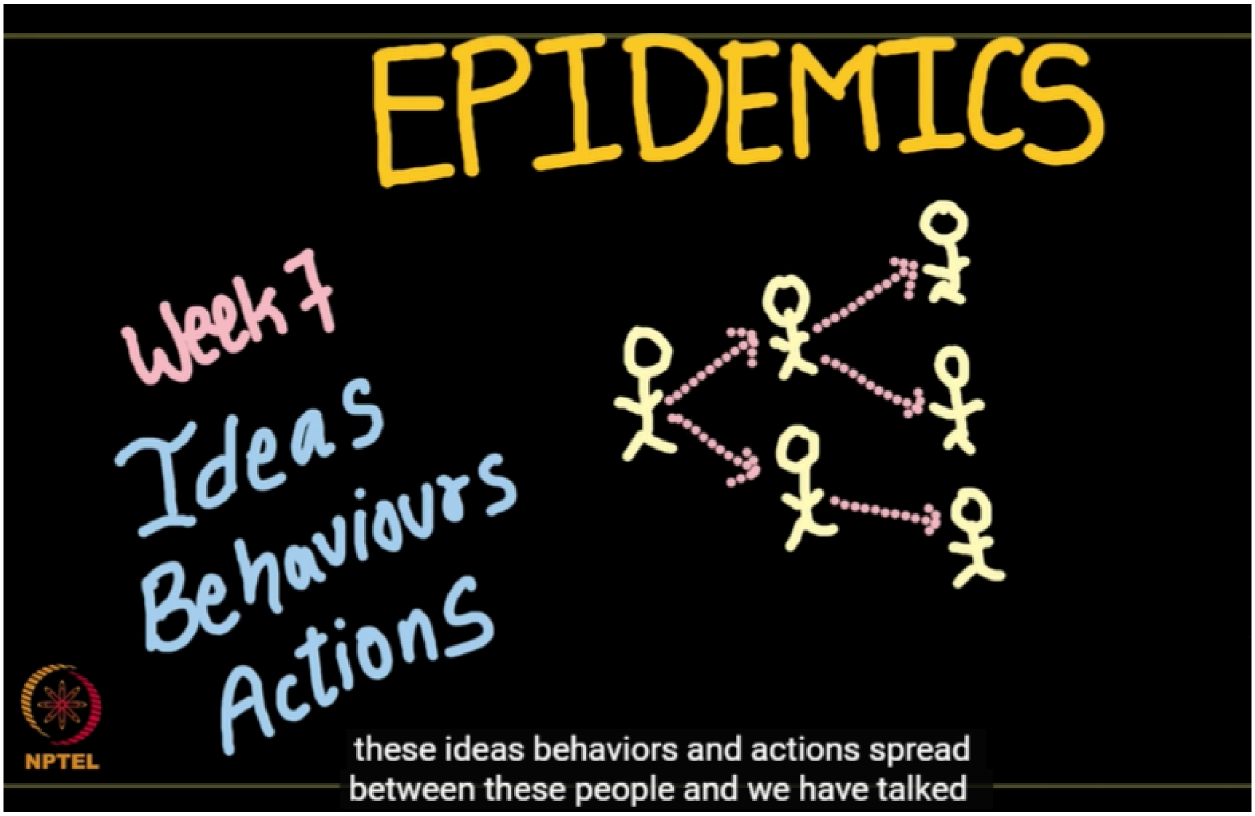
Long Tail is the long extension that continues and continues for the unpopular words

Long tail phenomena is nothing but the fact

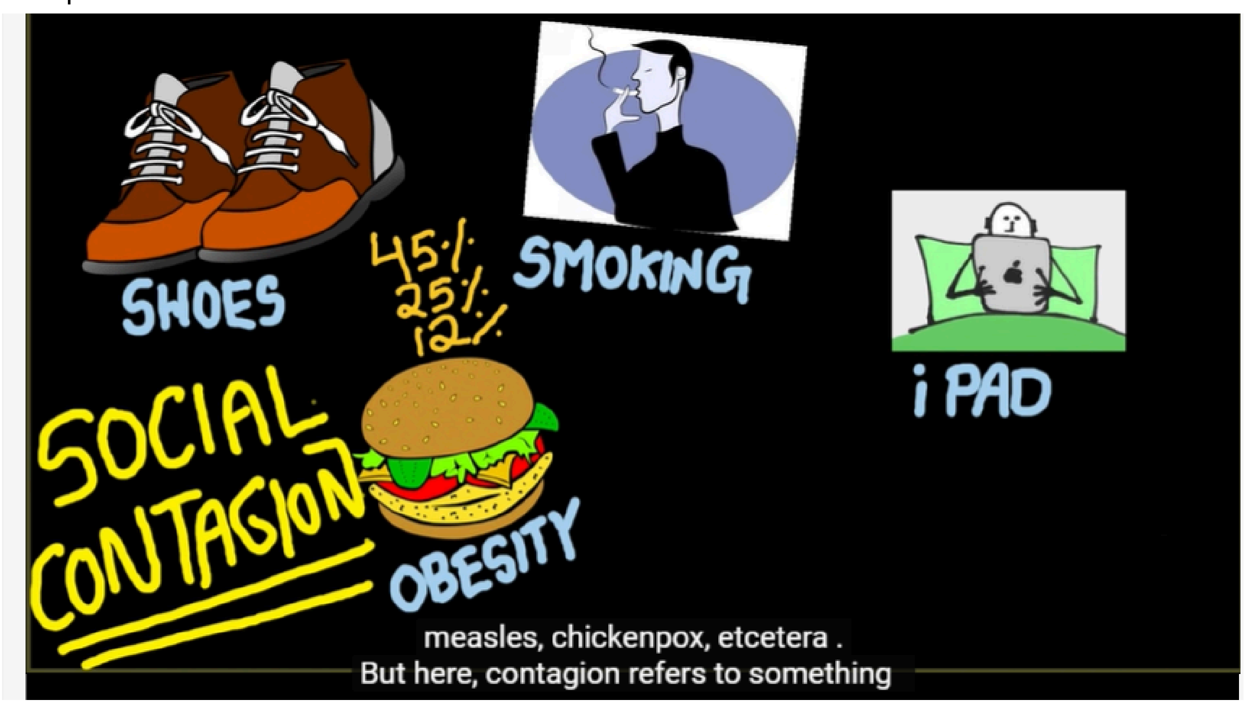
that; um so, the the the popular products areless in number although they are popular and less

popular products are more in number although they are less popular the sort of dominate thespace .

Lecture 128 - Epidemics- An Introduction



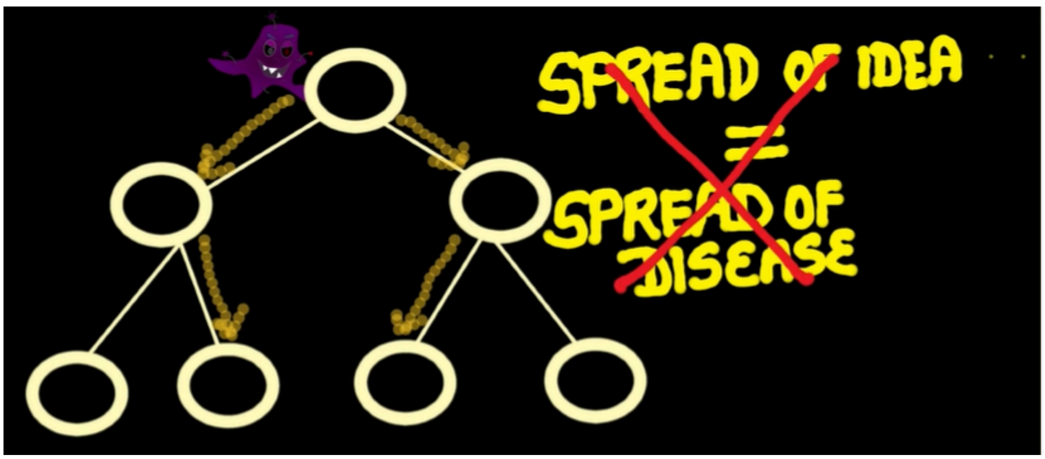
Examples we studied -



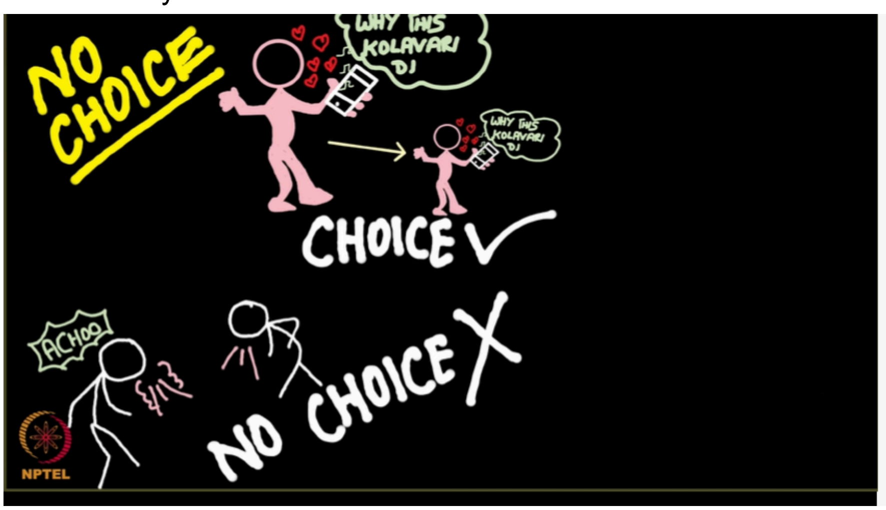
If your frnd is obese, ur chance of obesity is 45prcnt. If his frnd is obsese, ur chance is 25prcnt, futher if his frnd then 12prcnt. After this there is no correlation

need to be controlled, we we cannot afford
to have epidemics. World has suffered from

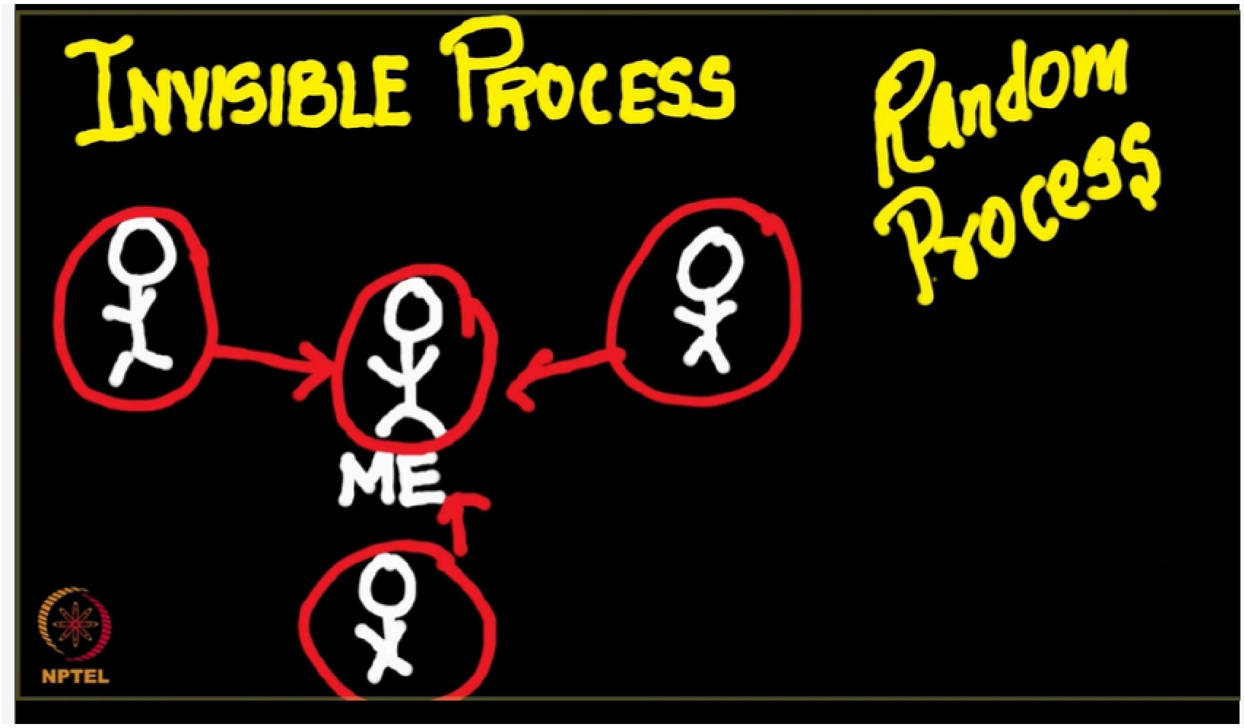
But dont u think this is a concern of biologist. Is it useful for us tu study? YES



Looks like Disease spreads similar to a disease, but actually its not.
Reasons why not same - 1. NO CHOICE



2. INVISIBLE PROCESS - spreading of disease is an invisible process as u dont know from where u got the disease. It is a random process, we need a different model to study it



Lecture 129 - Introduction to epidemics (contd..)
2 factors mattering for spreading of the disease or idea - PATHOGEN & NETWORK

1. the flu itself. So, what is spreading on this network is important. For example,

its degree of contagiousness matter, how contagious this flu is. Taking an analogy with

these spreading of an idea as we have seen, something like a piece of code spreads less

quickly as compared to a juicy piece of gossip. Similarly in the case of diseases there are

certain diseases which is spread more quickly as compared to other diseases.

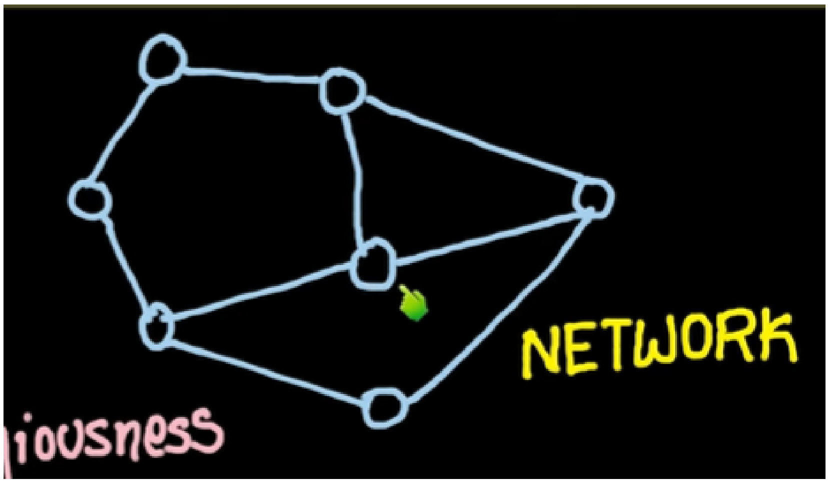
Measles and flu spread quite quickly then ebola and HIV.

2. if you look at the um network which is shown to in this figure, you can see that there are

quite a less number of edges in this network. The network is less dense rather we call that ntwrk is sparse

this network is So, you put any disease over there it will slowly move to this network;

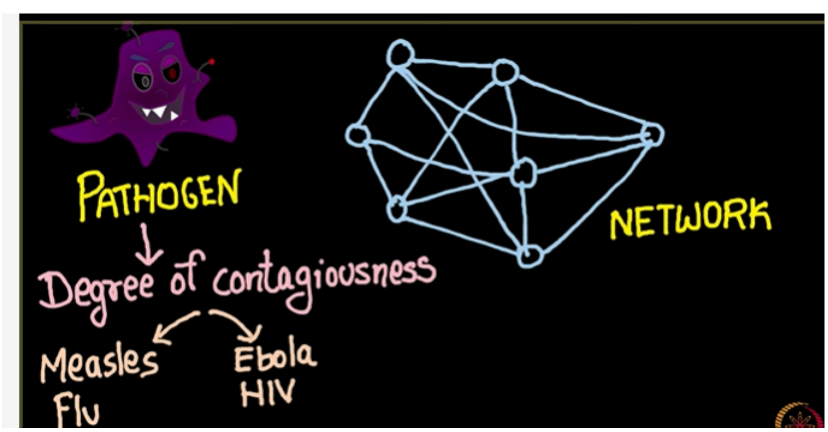
rather it can die away quickly, but if your network is say something like this and I put



IF a lot of edges there between these people, so you put any disease on this network and this

disease willthis disease will quickly spread on this network because, your network is dense.

So, yes one thing is the density of the network or let us say the sparcity of the network it



interesting to note here is do you see that this
pathogen has something to do with the network? So,

So, the network is going to be very
dense in the case of such a contagion, if the contagion is a flu, b

Bcz it can spread very quickly even by a keyboard

in the modeling of a HIV will

with the help of sexual contacts there will be
quiet less number of edges which will bewhich

will be counted in this contact network.
So, the network in the case of HIV is going to be very sparse.

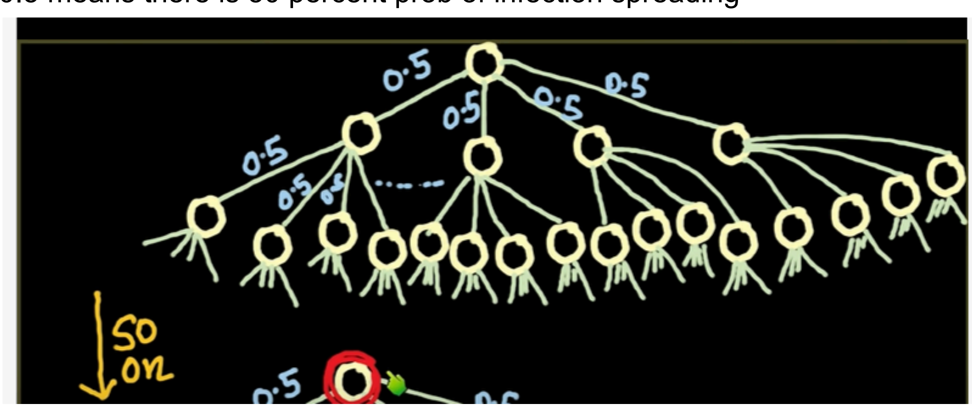
to be very sparse. So, whether your network is
dense or your network is sparse also depends

upon what kind of a pathogen we are talking
about. If this is the pathogen like flu then very dense, if pathogen = HIV then sparse

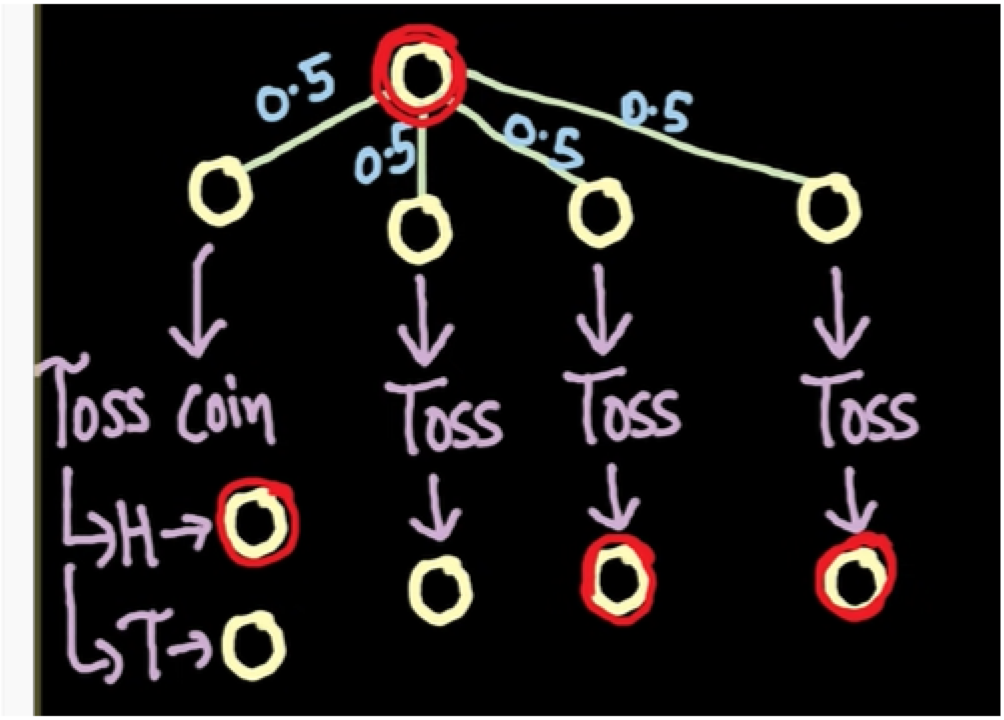
Blue – FLU
Orange - HIV

Lecture 130 - Simple Branching Process for Modeling Epidemics

Eg - taking a tree network example- one node having 4 children
0.5 means there is 50 percent prob of infection spreading



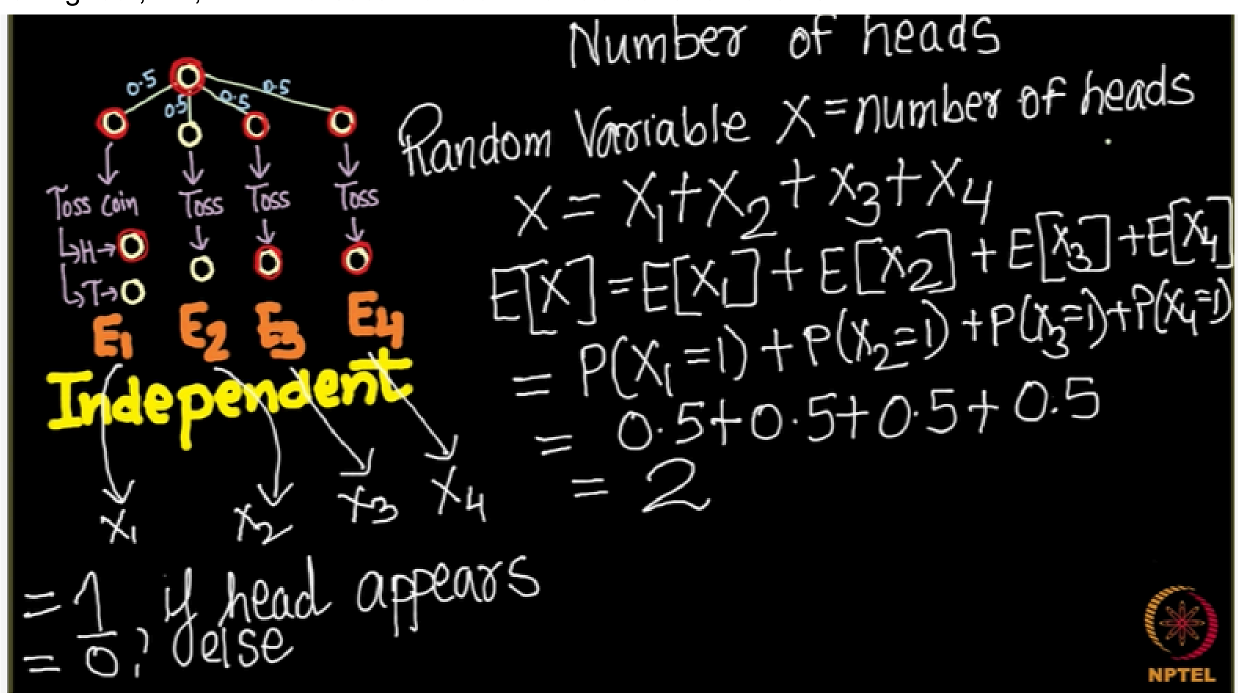
If red is infected, how many children can it infect?



=> 3 children are infected - 1,3,4

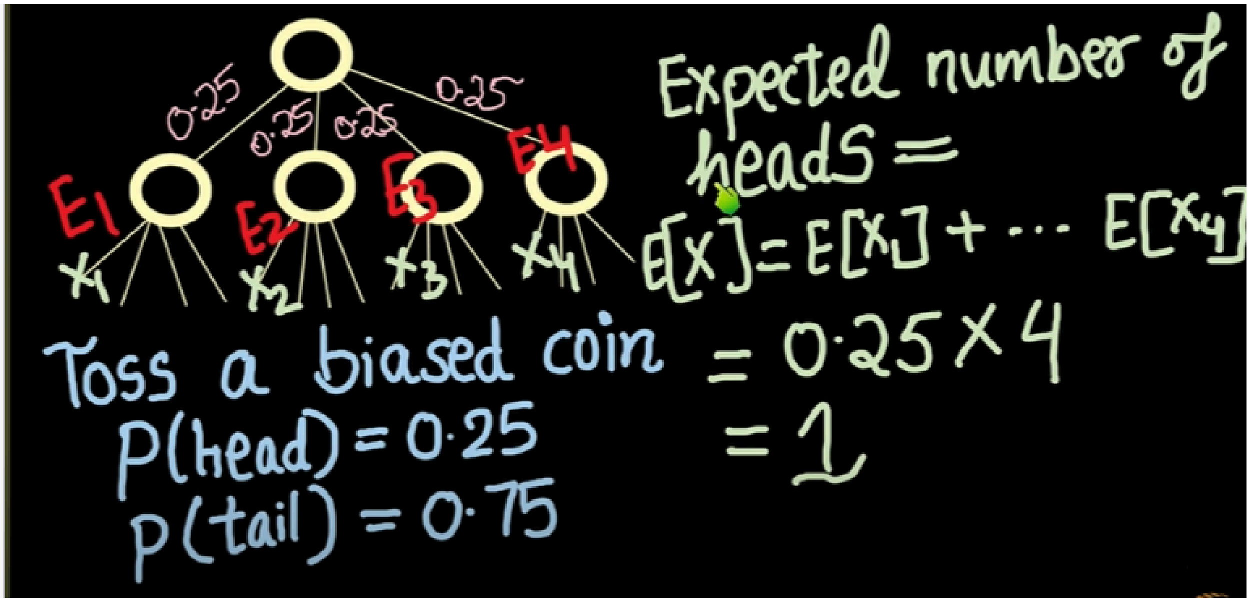
What happens on an average??

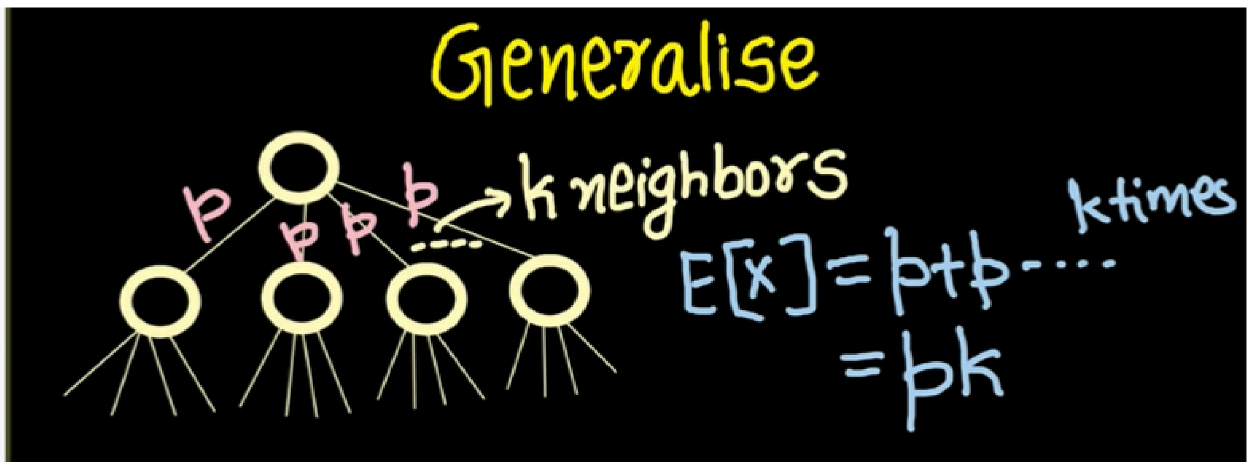
Taking four, x1 ,x2 … **indicator random variables** for all events



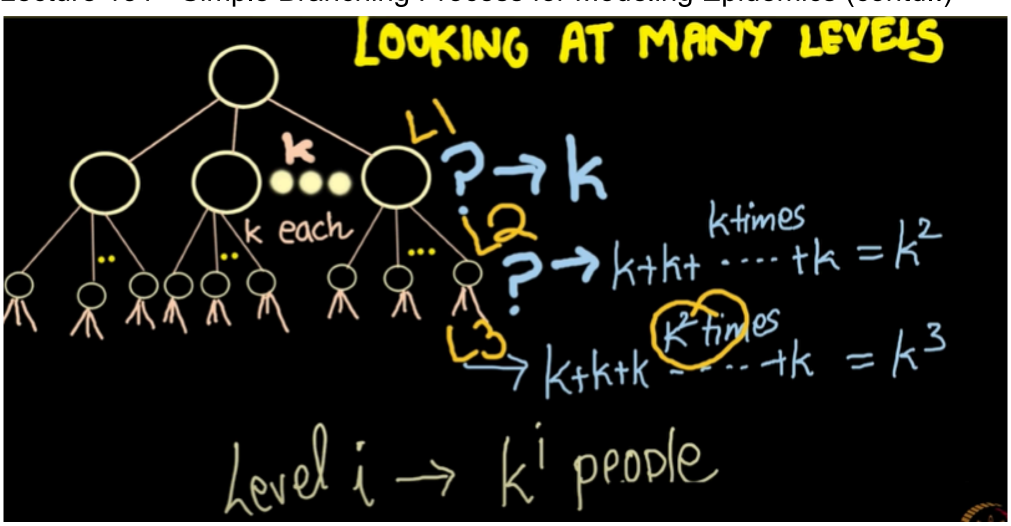
Or simply **0.5 x 4 = 2**

Another example -

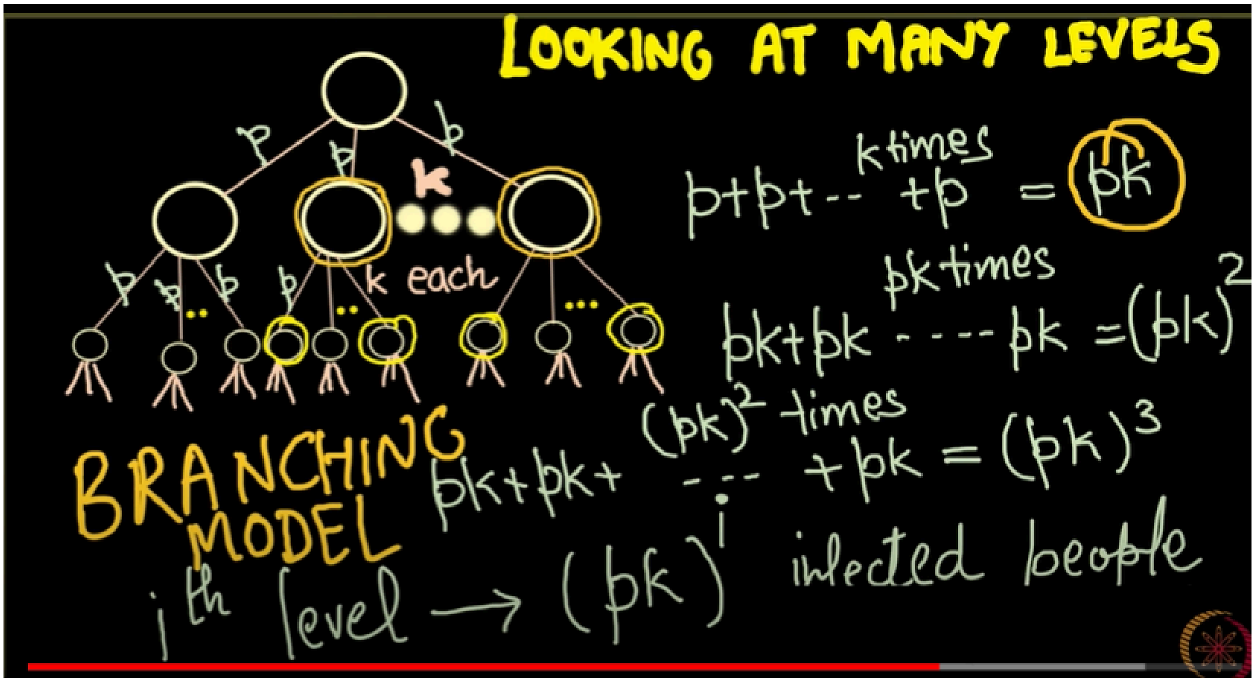


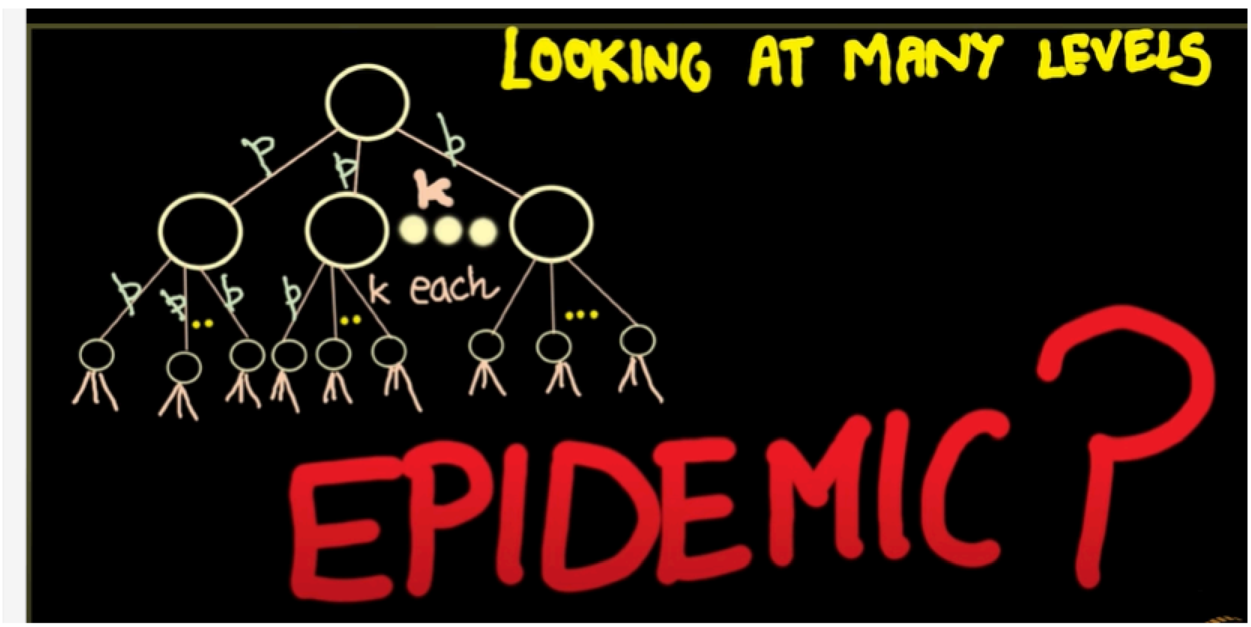


Lecture 131 - Simple Branching Process for Modeling Epidemics (contd..)



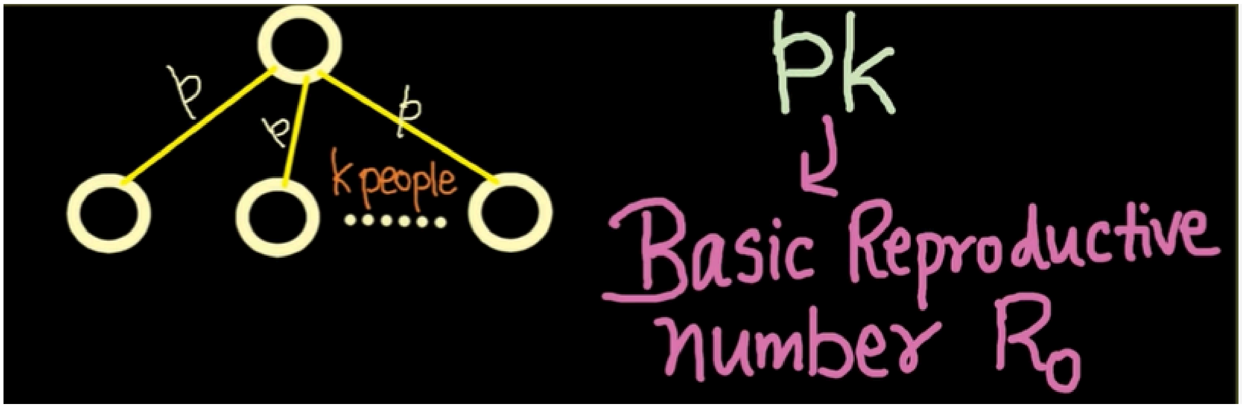
BRANCHING MODEL

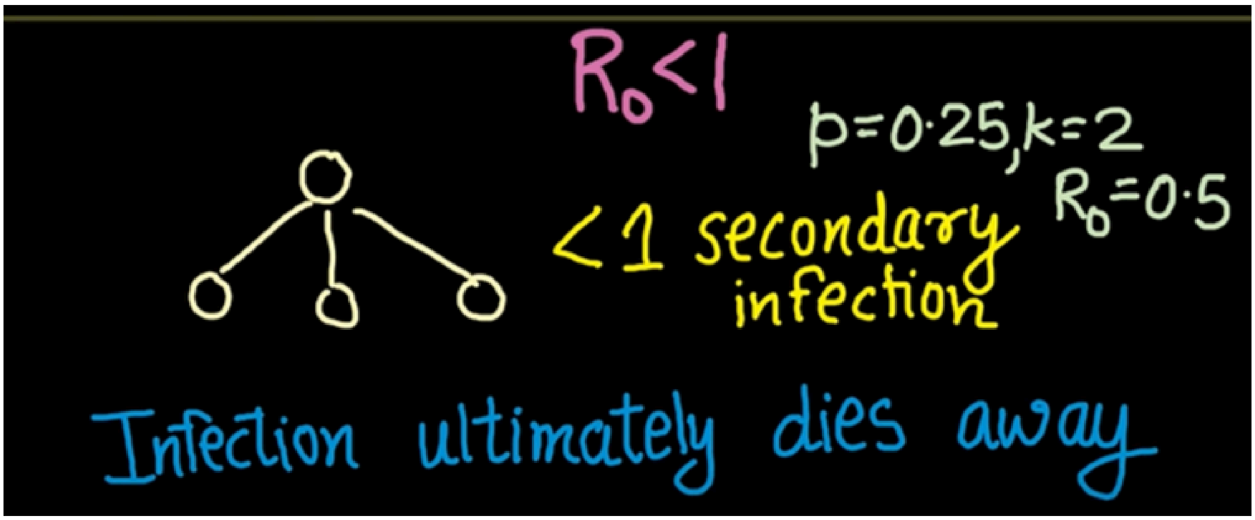




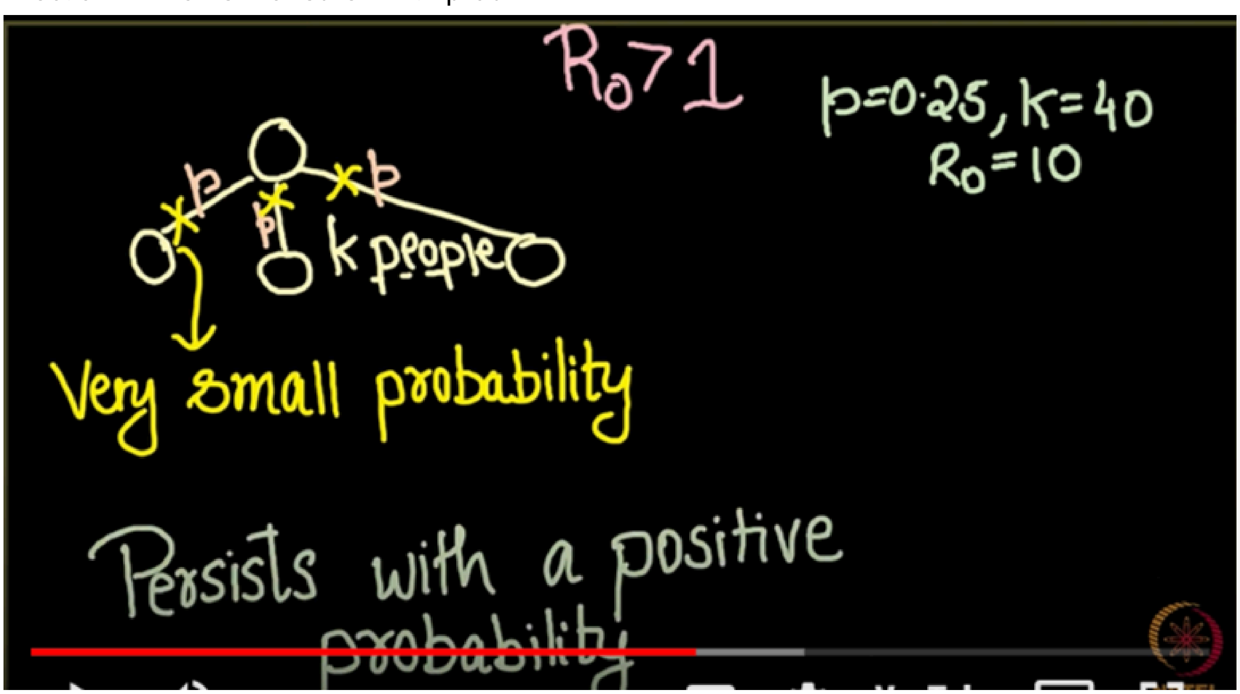
We see if this disease spread to infinte level, if yes, then it has become epidemic .
Next lec, by looking at p and k, we can predict whether it is going to become epidemic or not.

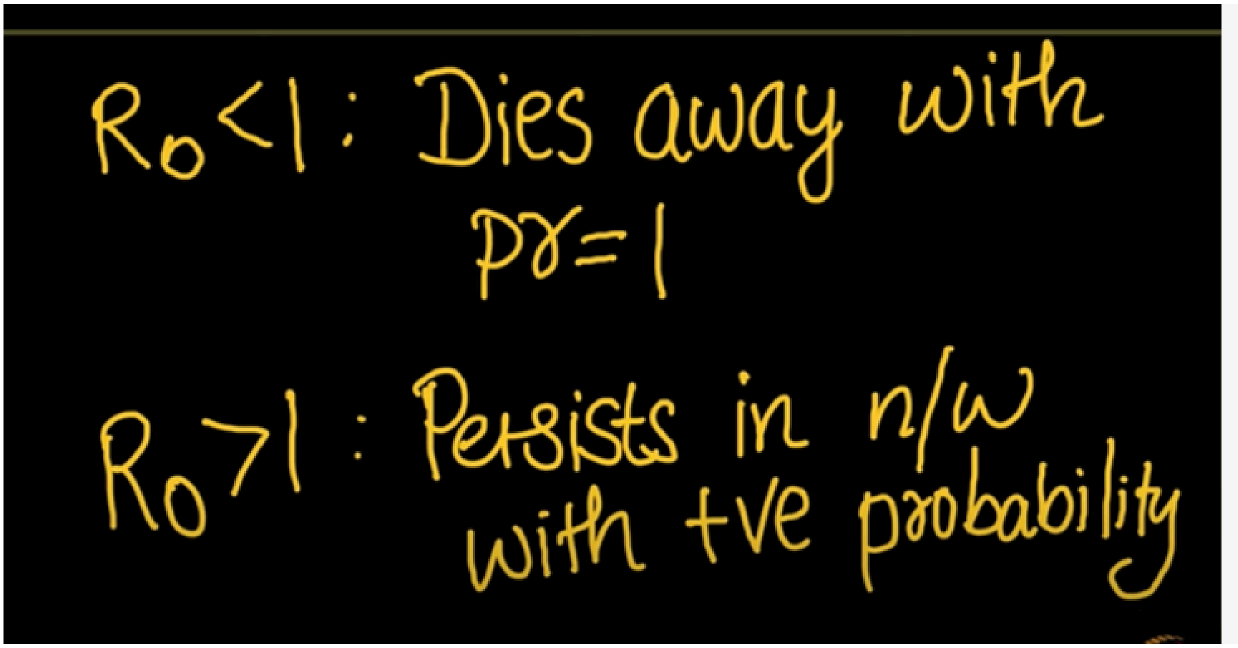
Lecture 132- Basic reproductive number



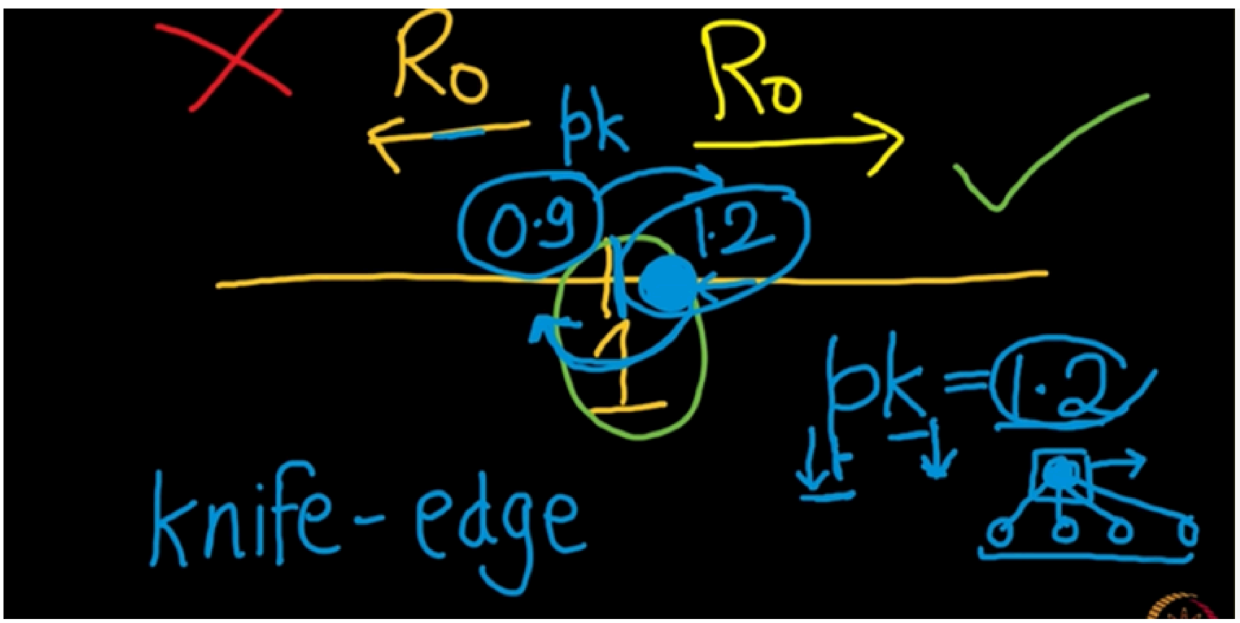


Infection will vanish for sure - with prob = 1



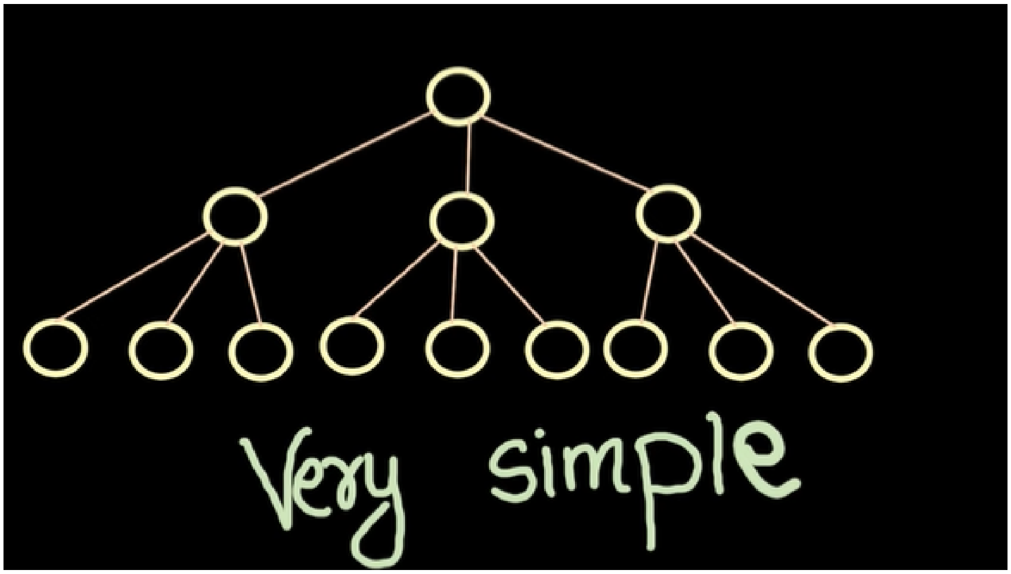


In 2nd case, it can also die if all links fail **but this prob is very very less.**

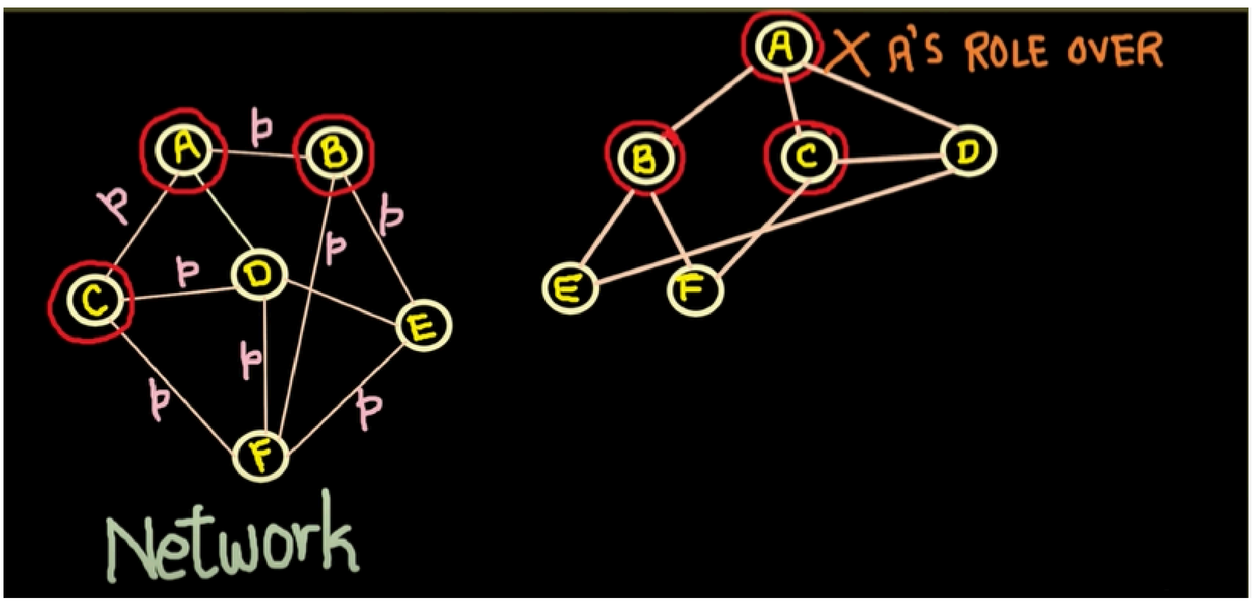


It is easy to make an epidemic (1.2) to die off by reducinhg k and p
KNIFE EDGE PROPERTY - > a minor change in p or k can result in epidemic or make an epidemic die off

Lecture 133- Modeling epidemics on complex networks



Complicated network - NOT A TREE

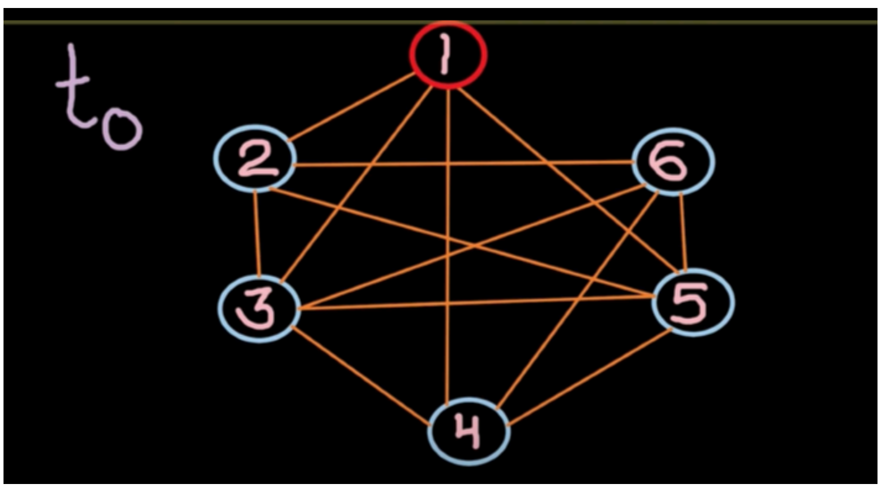


SIR EPIDEMIC MODEL

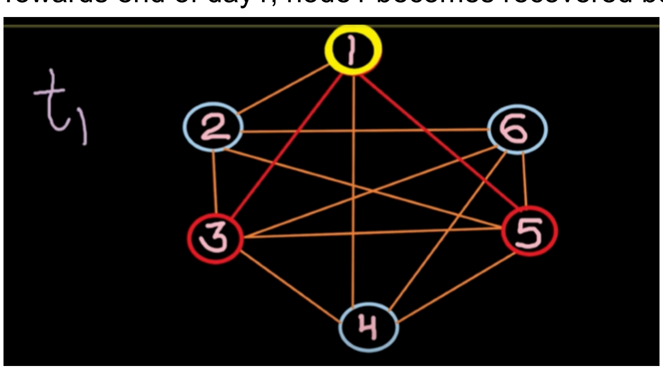
Either u wil be killed by the disease(removed by the network)
OR u will develop a permanent immunity against the disease (RECOVERED)
U cant go from infected -> susceptible(can catch a disease only once)

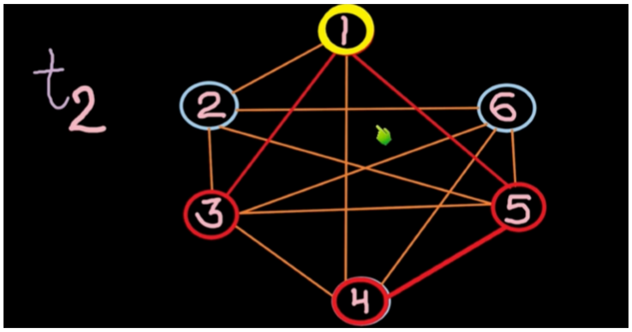
Sucsceptible Ti = time till which one is infected Removed

Lecture 134 - SIR and SIS spreading models
Consider Ti = 1
At t0, only Node 1 is infected

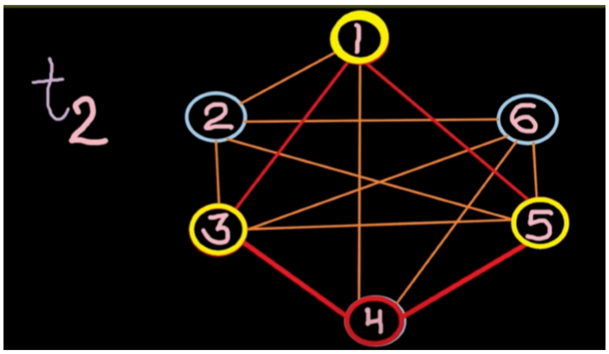


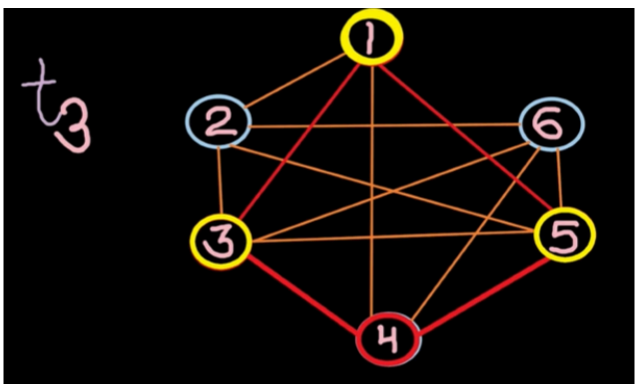
Infecting the neighbors are random. It can be different nodes at also at random .
Towards end of day1, node1 becomes recovered but infected node 3 and 5



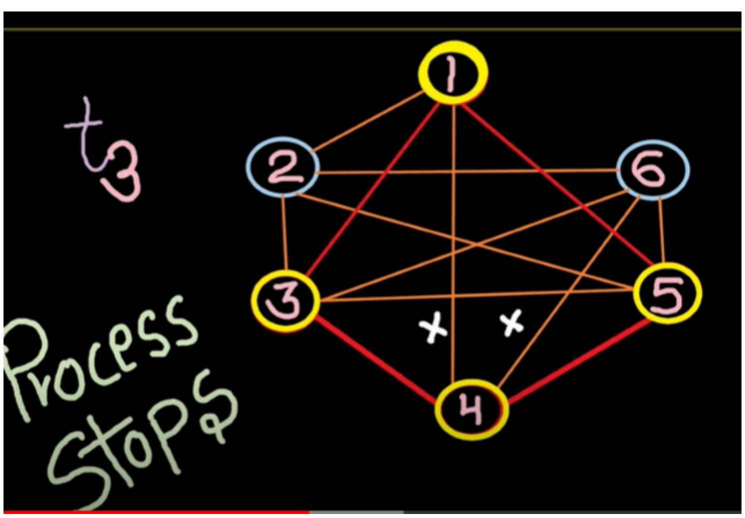


Towards end of day2, node 3 and 5 recover



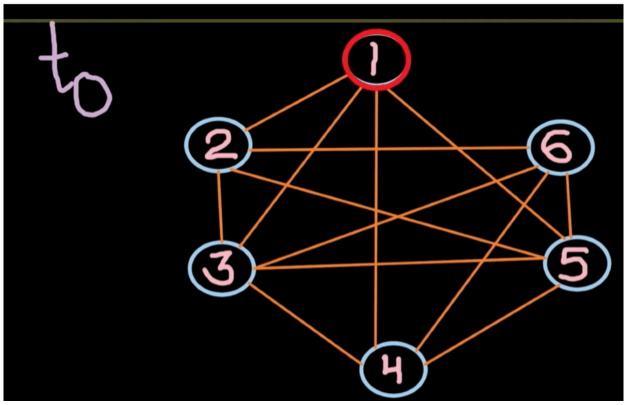


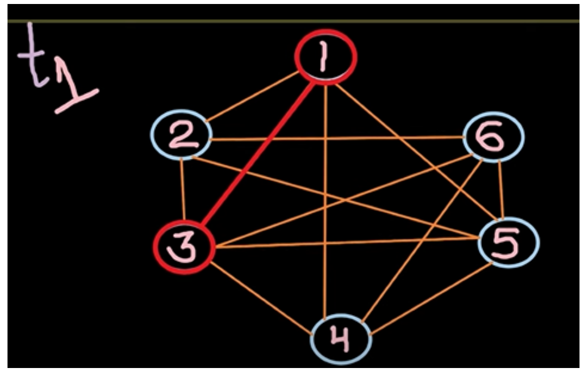
Node 4 couldnt infect anyone

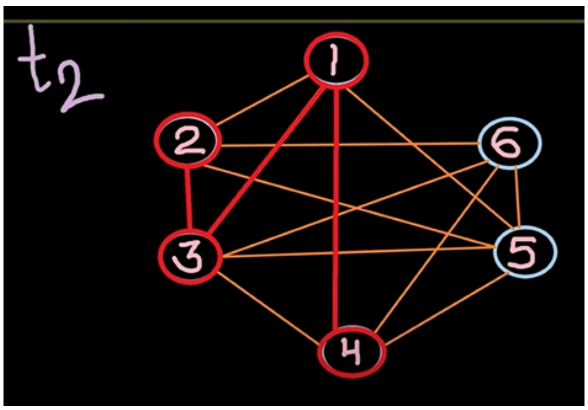


NO RED NODE AVAILABLE

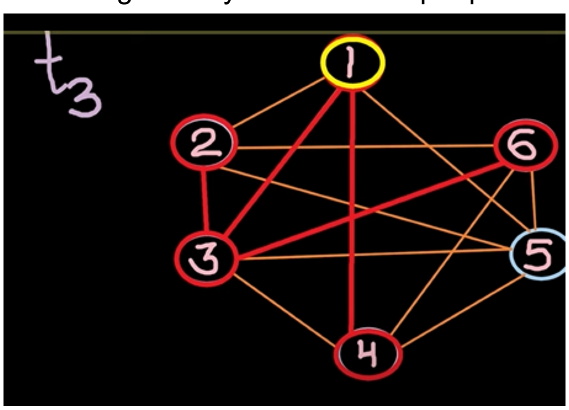
CONSIDER Ti = 2

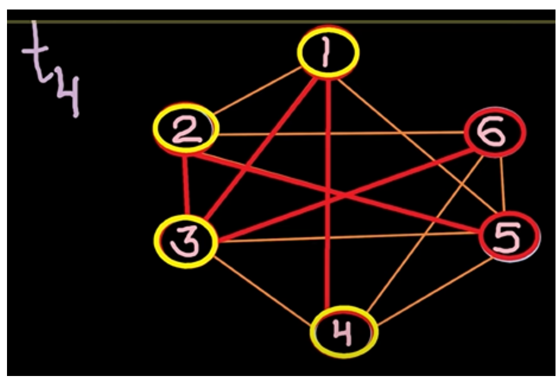


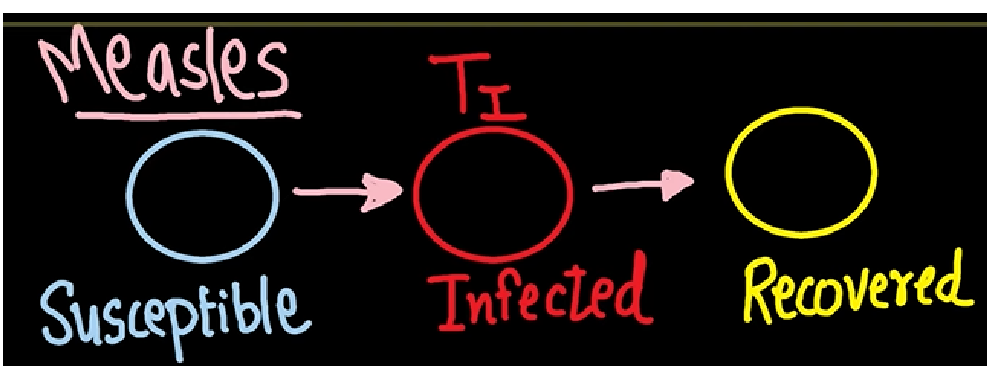




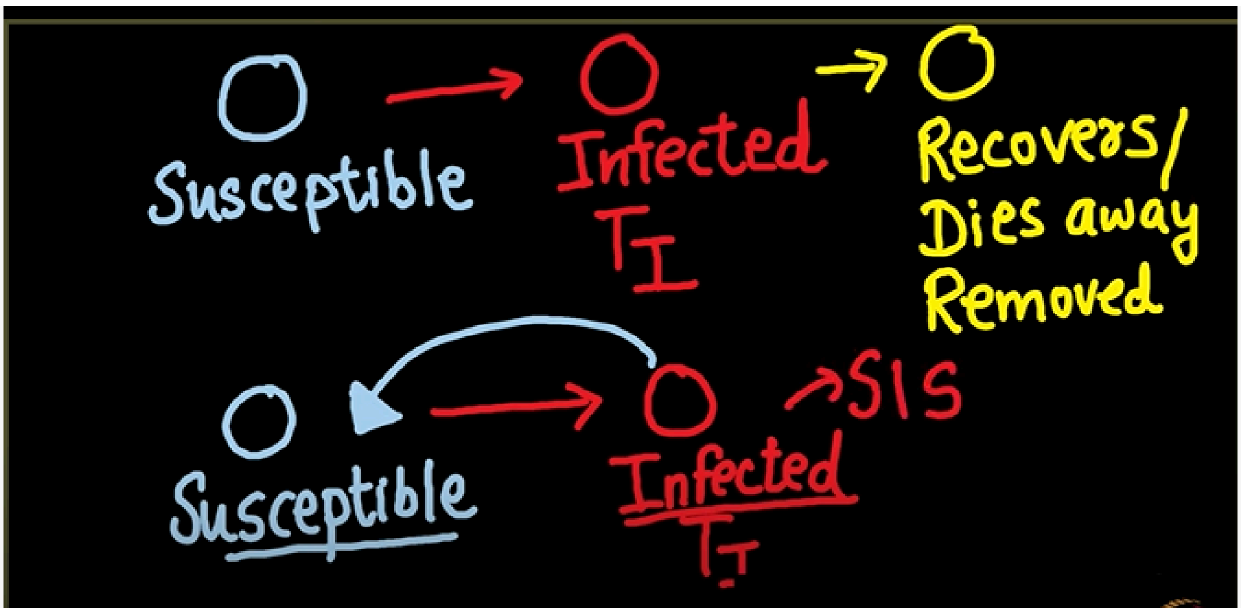
NODE 1 got 2 days to infect the people.

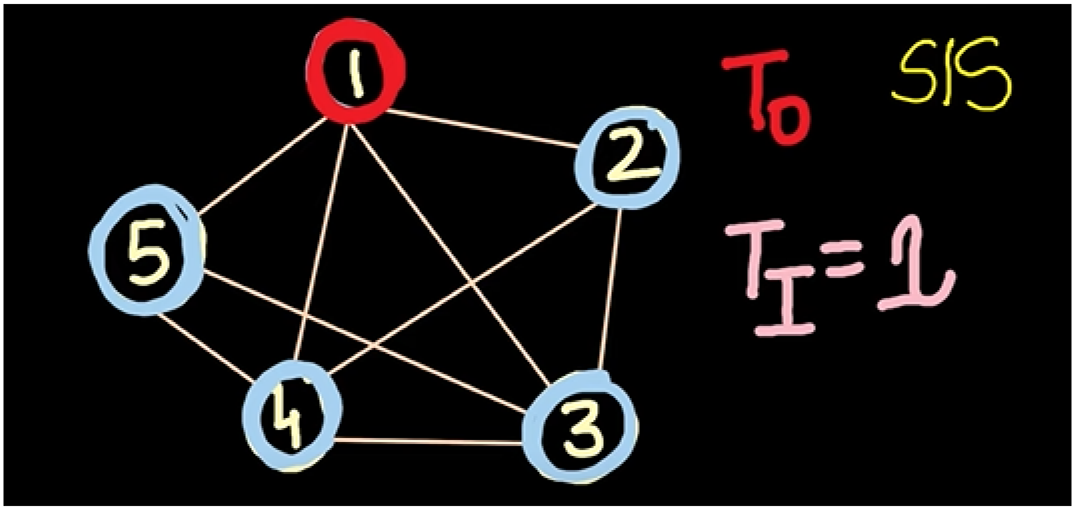


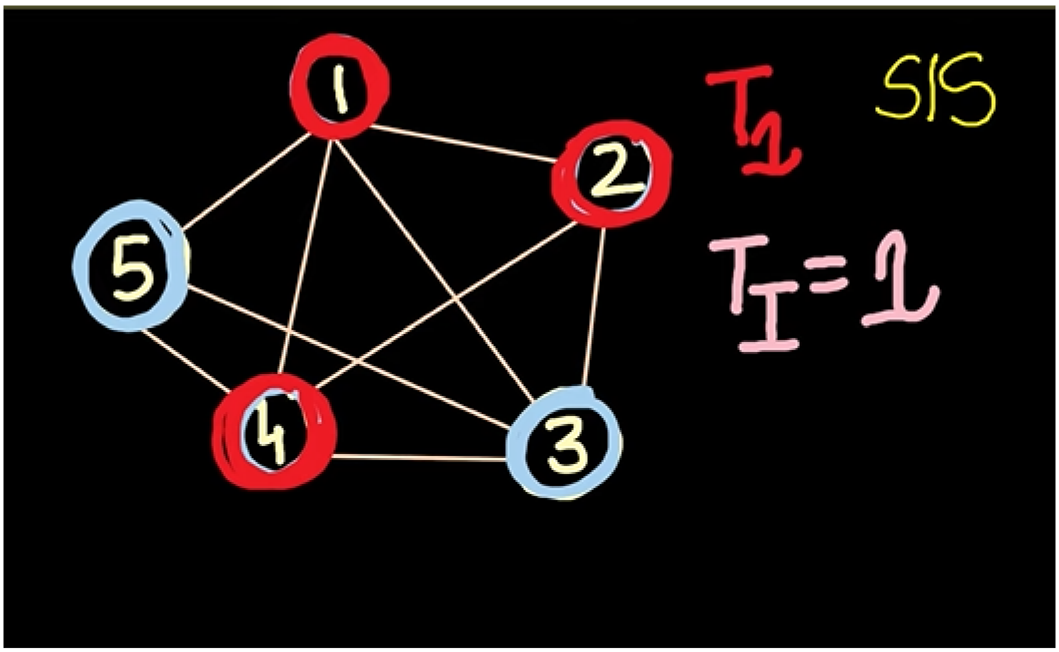




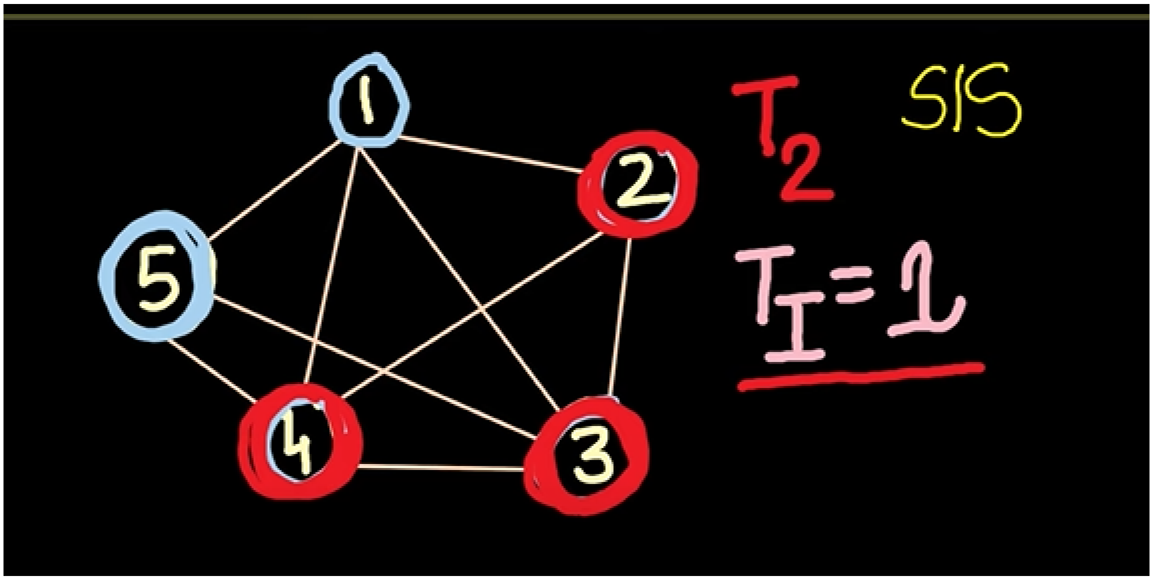
FOR COMMON COLD -

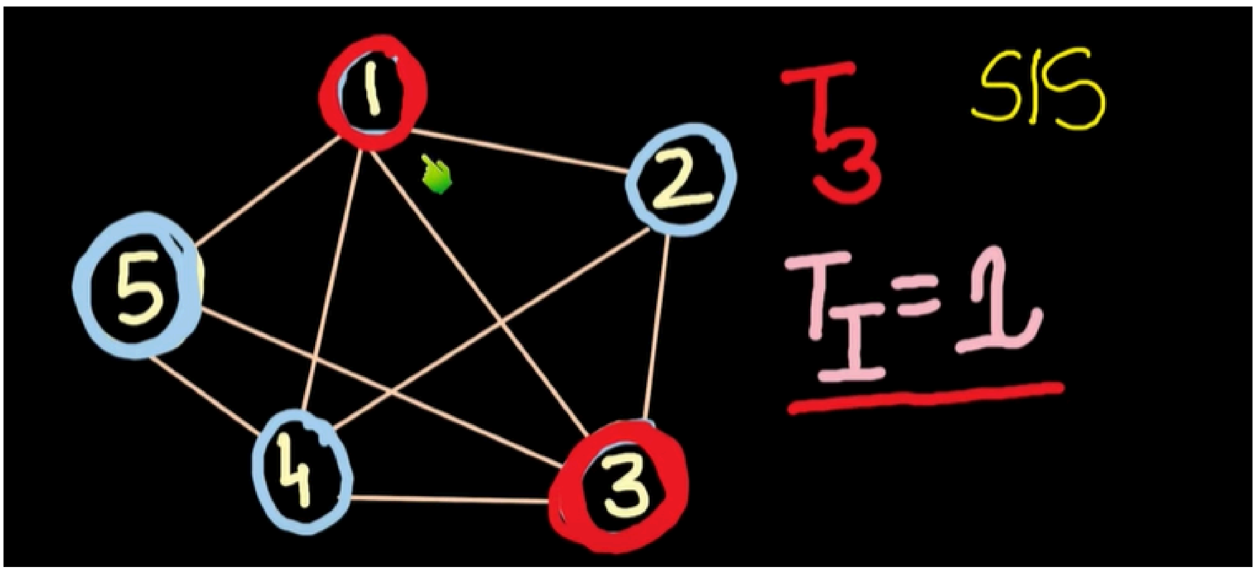




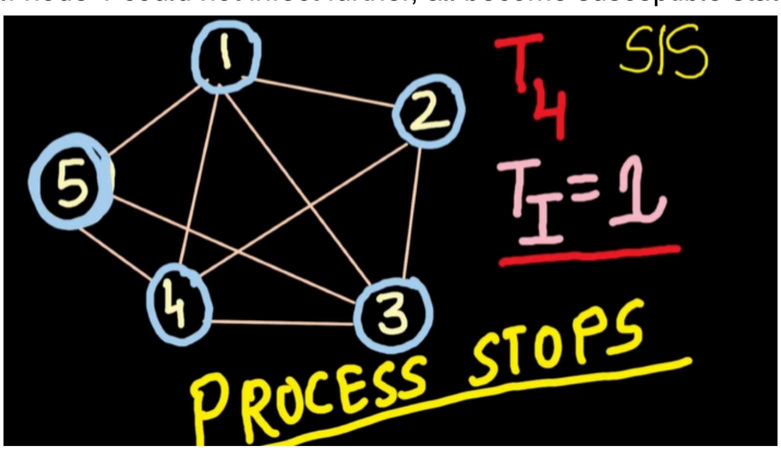


After recovering on day1, node 1 goes to susceptible state(not recovered)

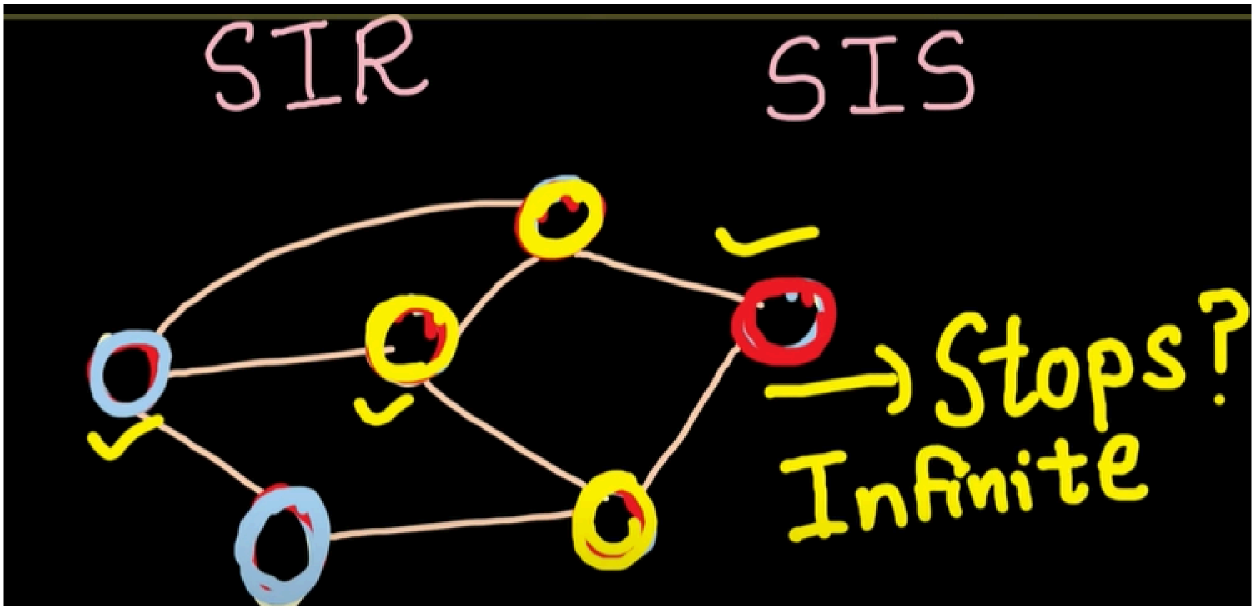




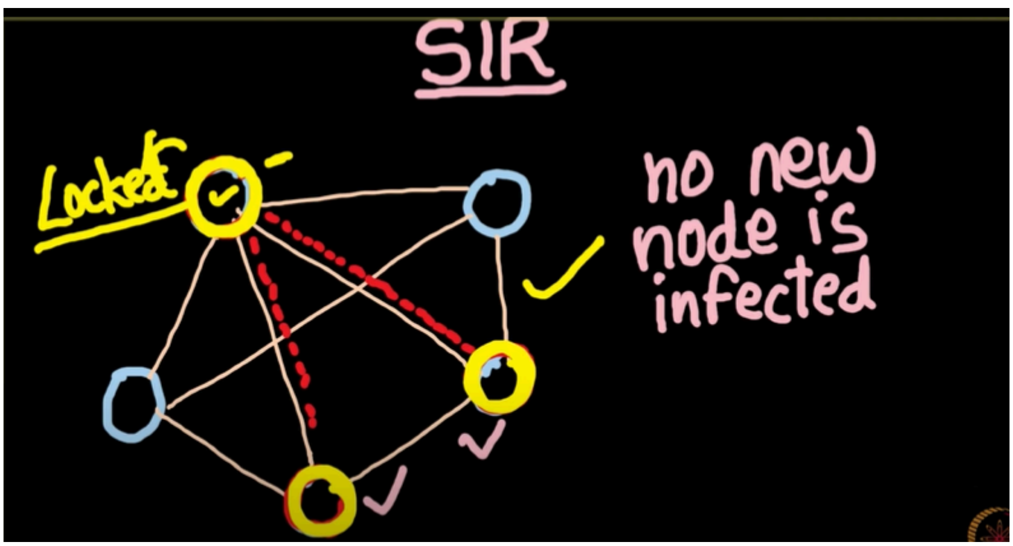
At t3, node1 is again infected.
If node 1 could not infect further, all become susceptible state, and process stops



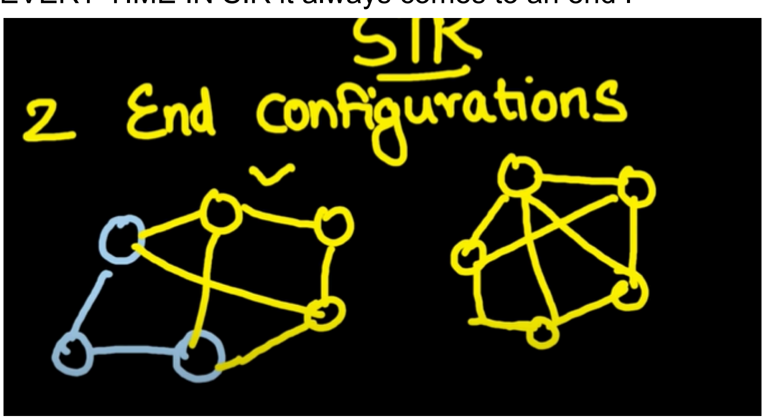
Lecture 135 - Comparison between SIR and SIS spreading models

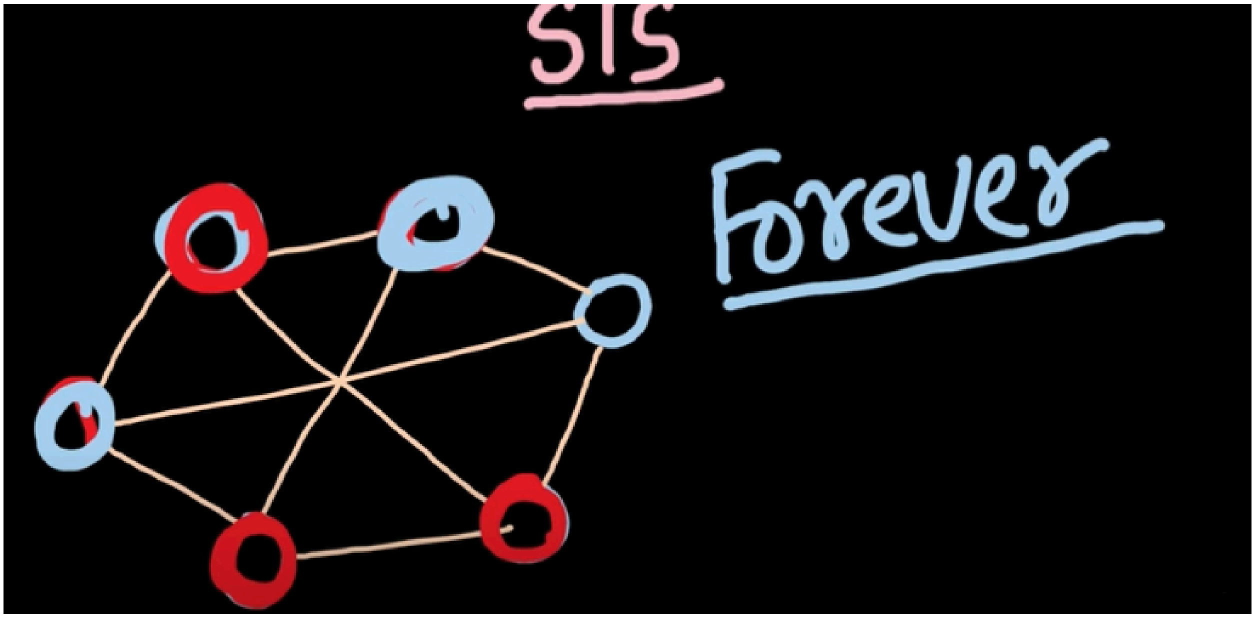


PROCESS STOPS -

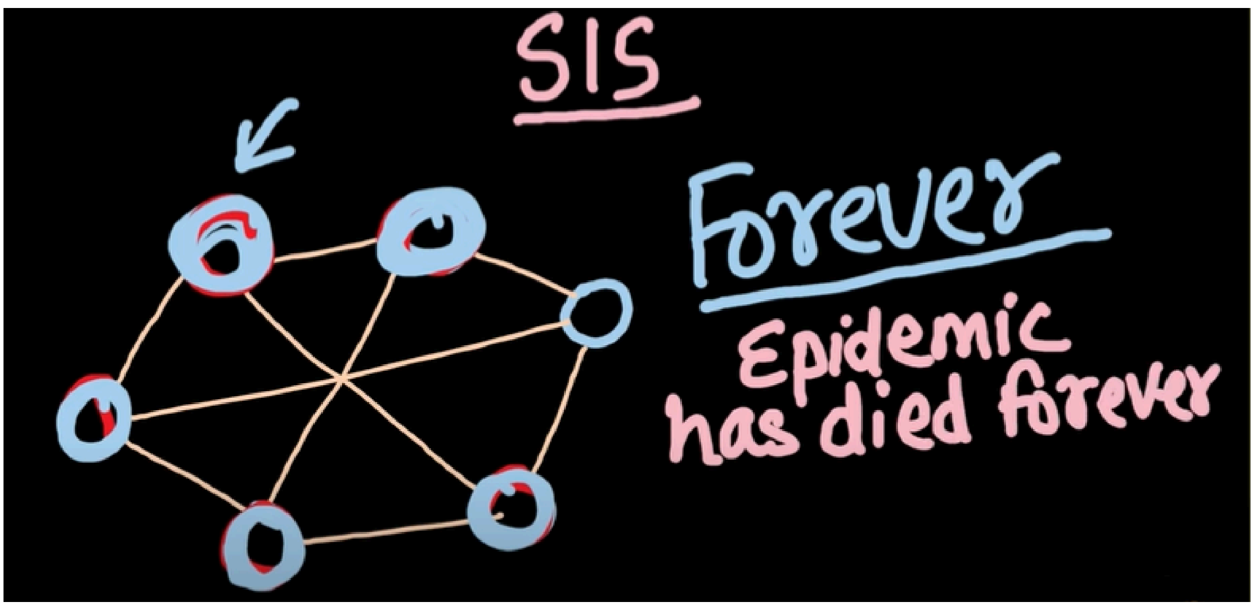


EVERY TIME IN SIR it always comes to an end .

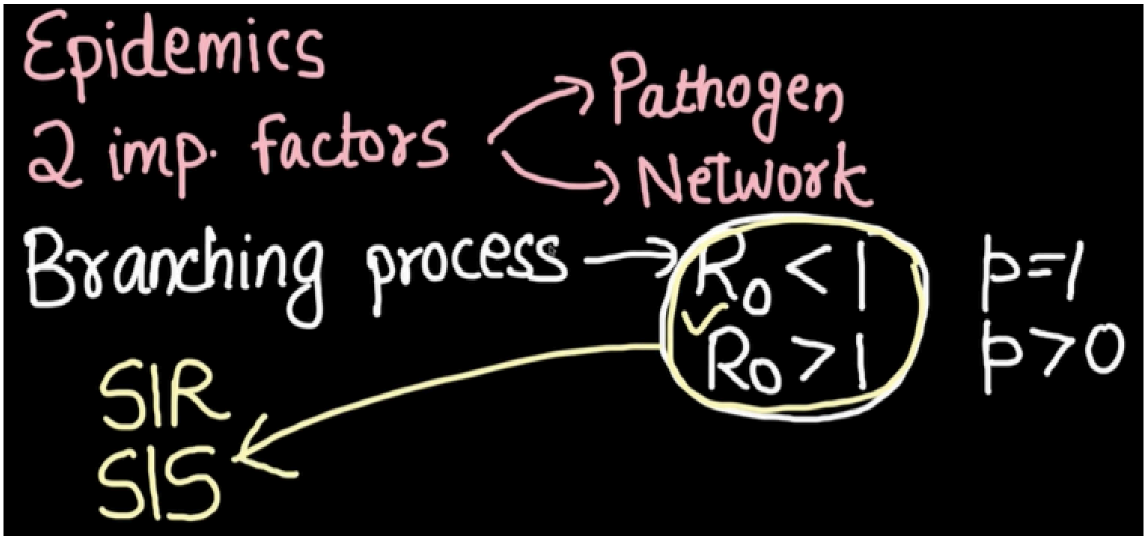




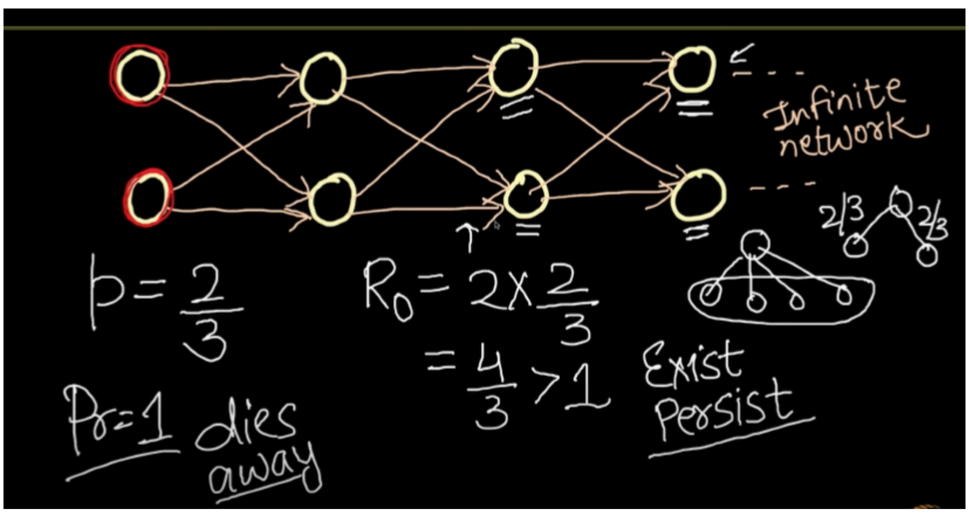
But in one case, all the nodes have become susceptible, then contagion dies forever

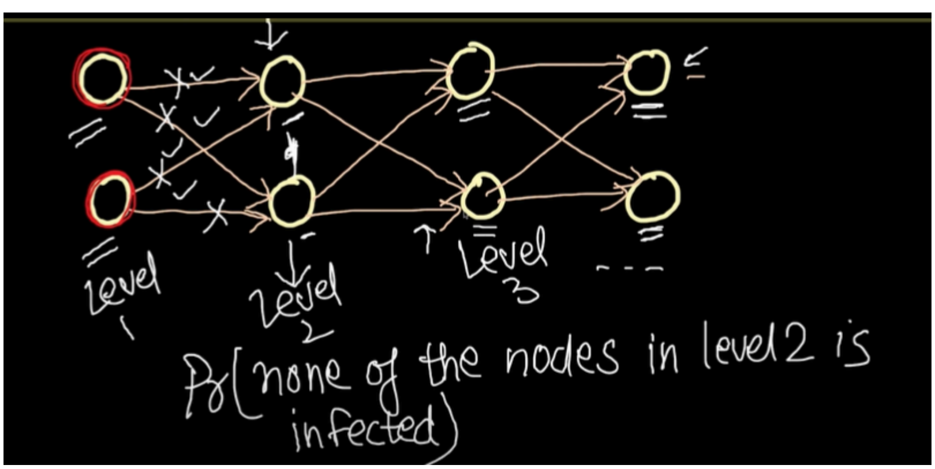


Lecture 136 - Basic Reproductive Number Revisited for Complex Networks

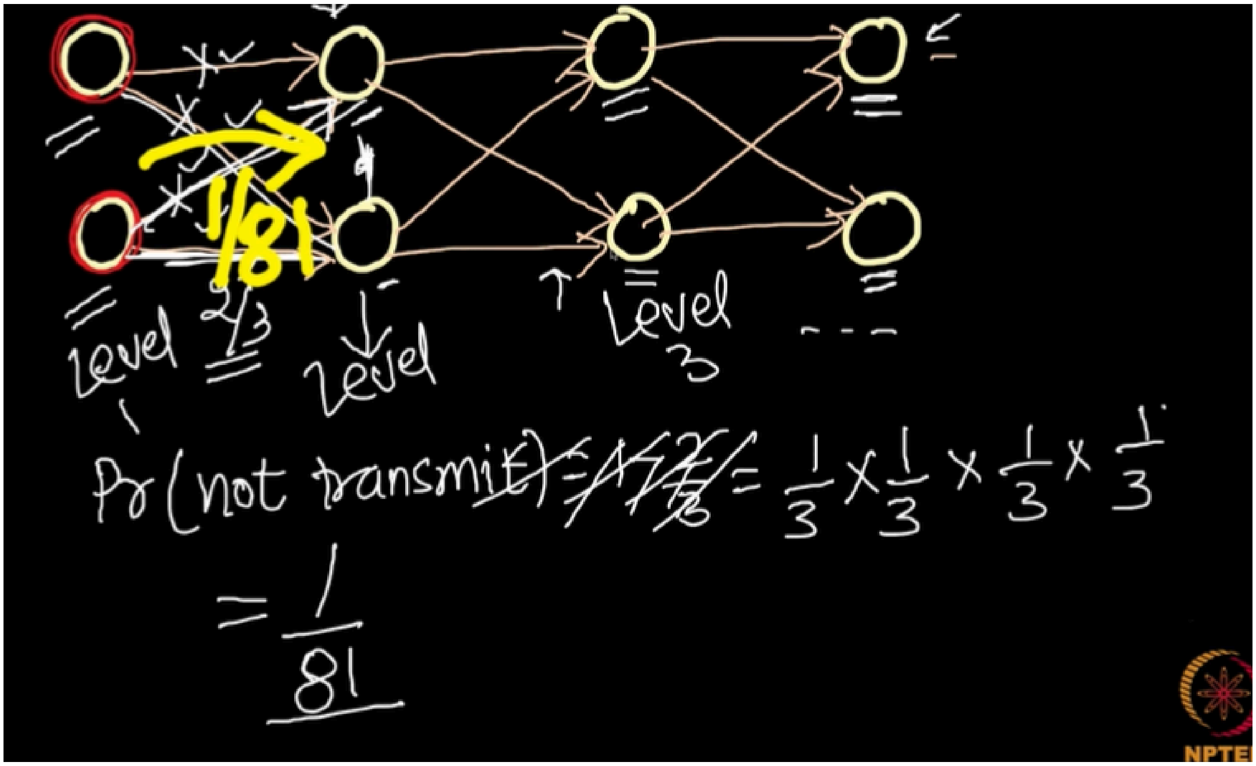


The theory of reproductive no. does not hold here in SIR and SIS
In sis model, if pr= 1 , then the disease dies away

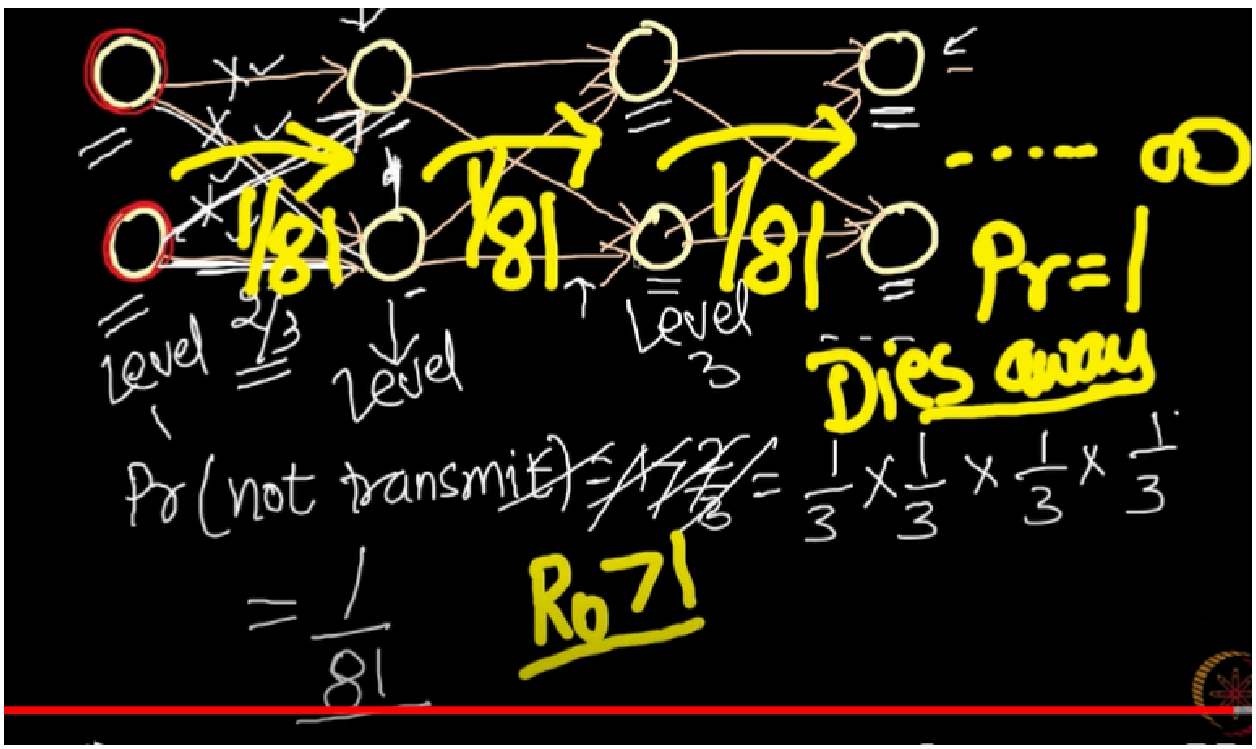




all links must fail, for infection to not transmit

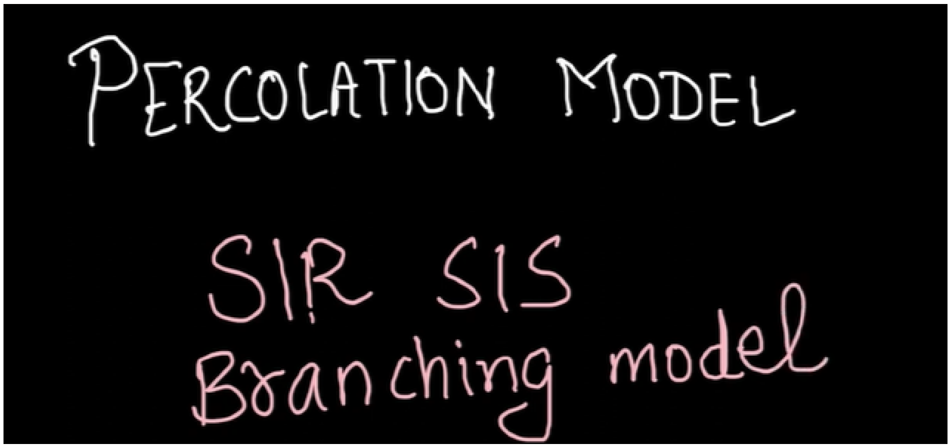


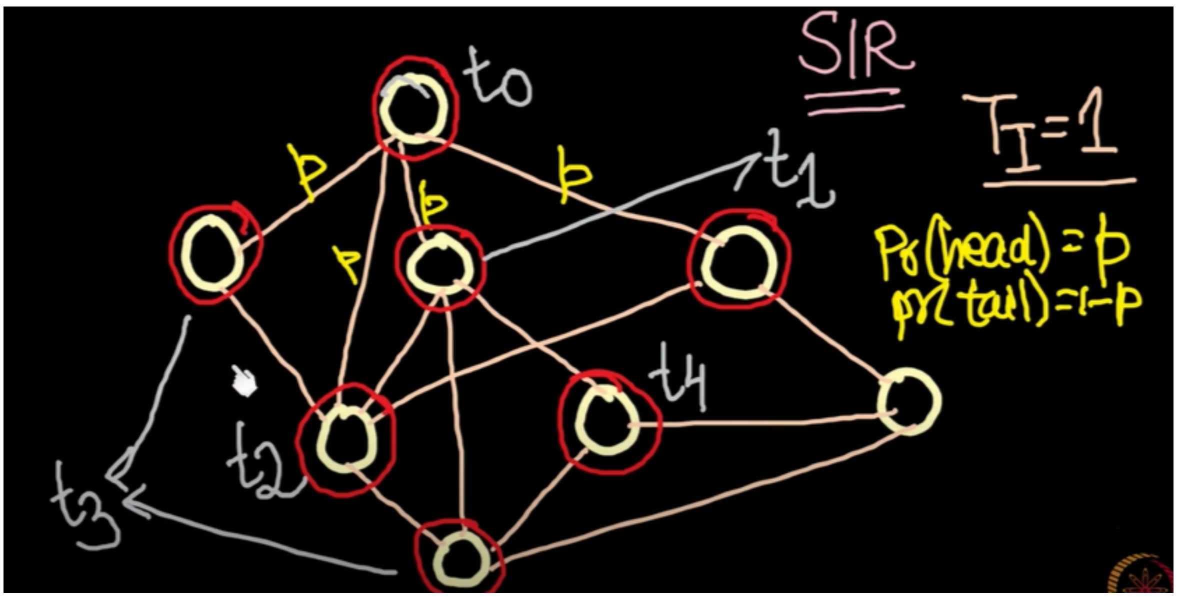
Disease dies with a prob of 1/81 at every level



Disease dies away with a probability of P=1 even if the R0 is greater than 1

Lecture 137 - Percolation model - **another way to look at already existing models**

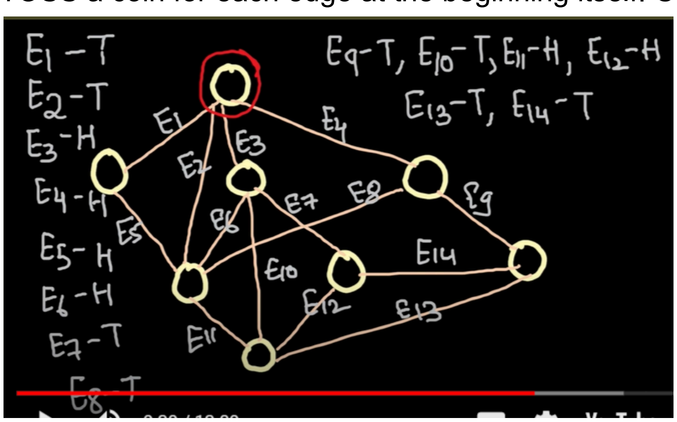




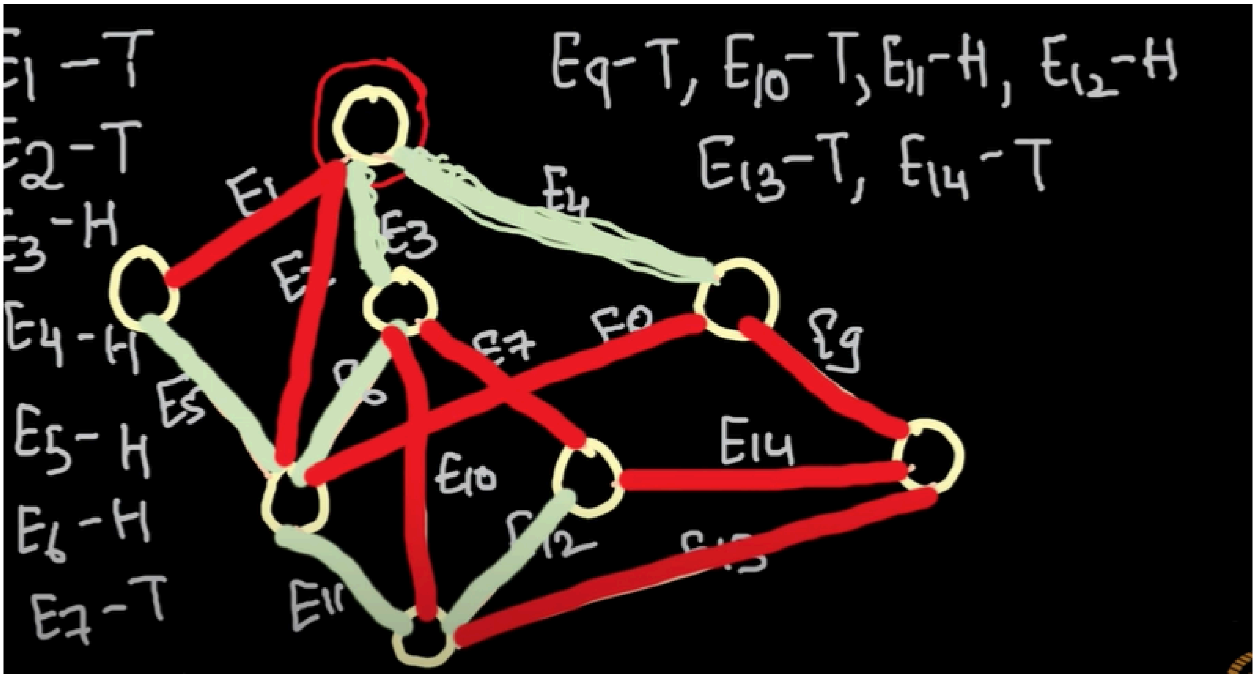
IF WE **REMOVE** THE NOTION OF **TIME** - MAKE EVERYTHING **STATIC** INSTEAD OF DYNAMIC

REMOVING THE TIME -

TOSS a coin for each edge at the beginning itself. Consider edges = pipe. Head = open pipe

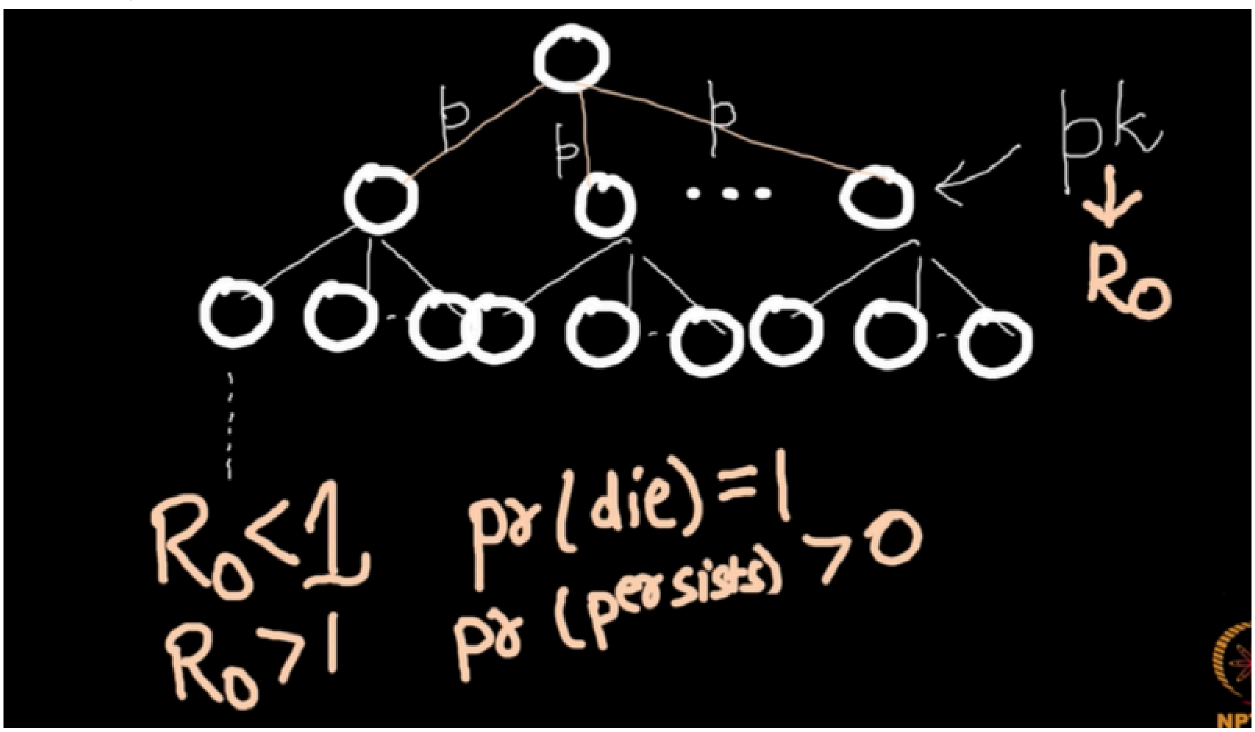


Red pipe = close, green pipe = open



Imagine water is flowing from the pipes. Check if water can reach a particular node from the pipes. Then it can be infected

Lecture 138 - Analysis of basic reproductive number in branching model (The problem statement)



WE WILL DO PROOF FOR THE ABOVE.

$q_n = P(\text{disease reaches level } n)$

$\quad = P(\text{disease persists level } n)$

$q^* = \lim_{n \to \infty} q_n = 1$ (Epidemic)

$\rightarrow 0$ (Died away)

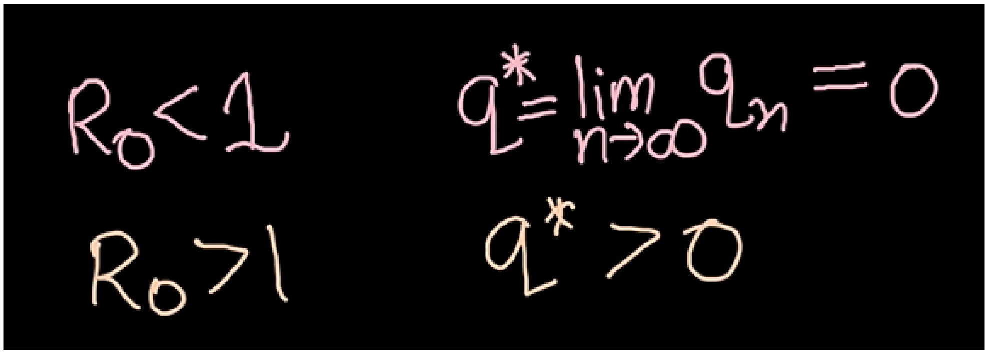
To Prove

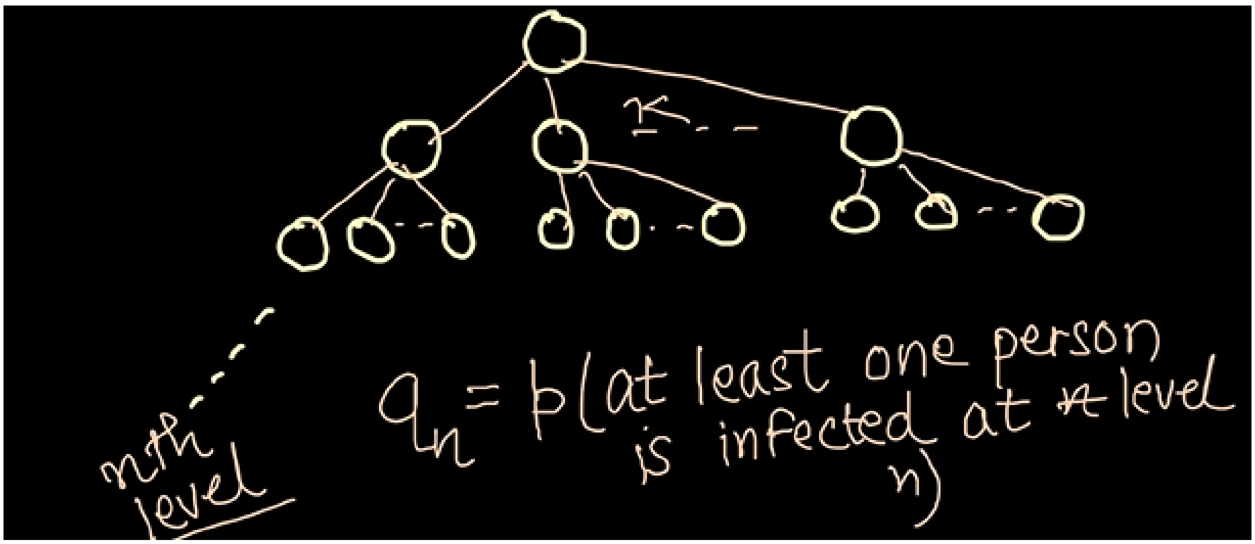
$R_0 < 1 \quad q^* = 0 \rightarrow$ for sure

$R_0 > 1 \quad q^* > 0 \rightarrow$ +ve persists

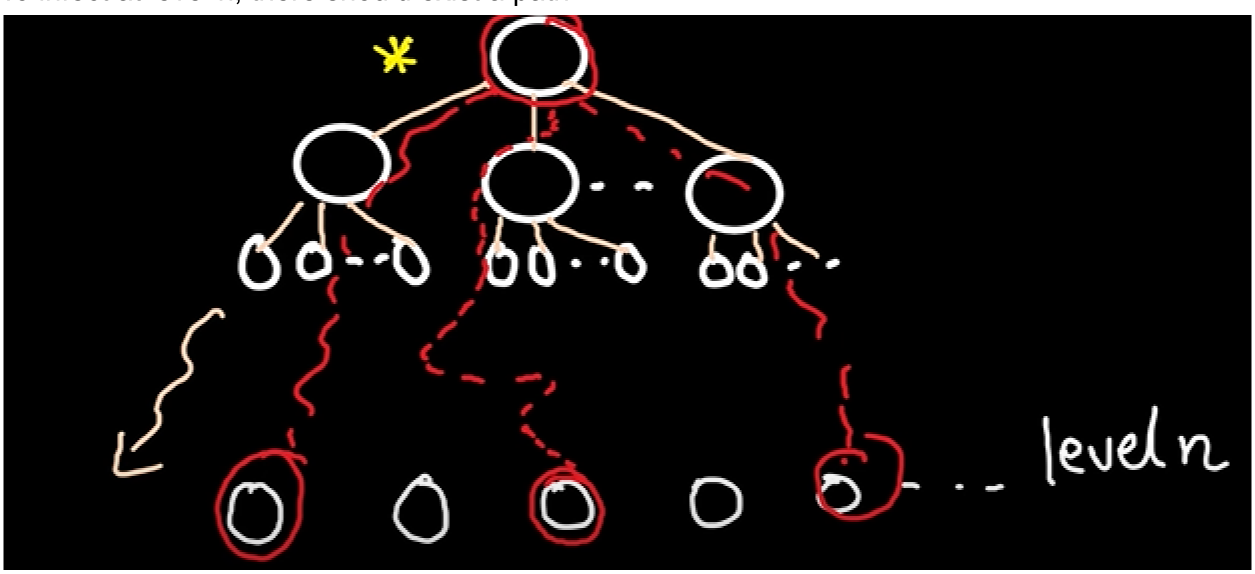
Lecture 139 - Analyzing basic reproductive number 2

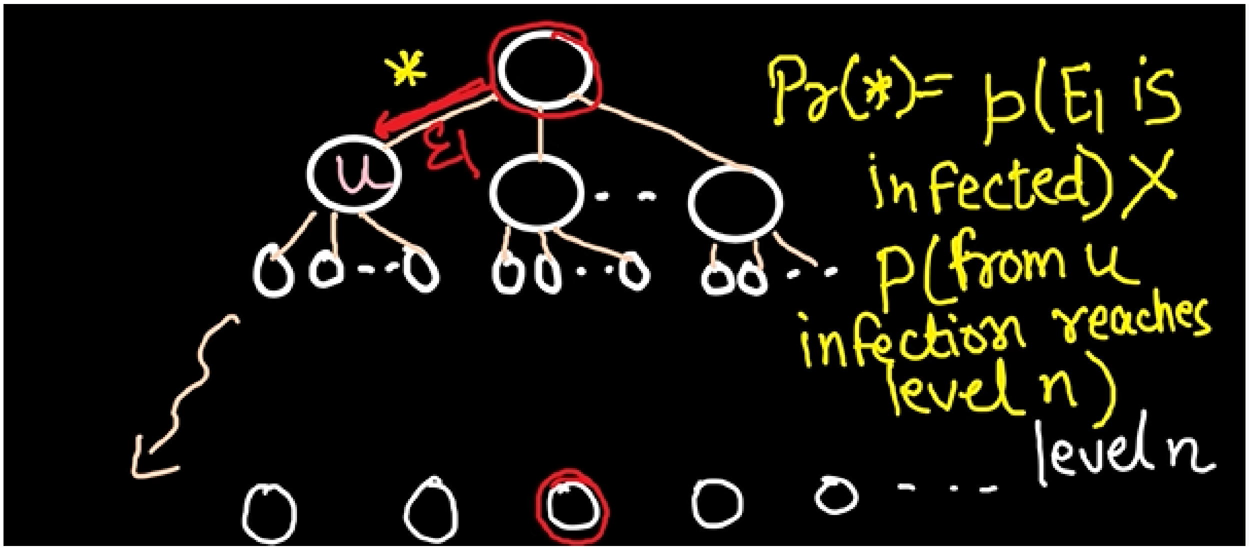
To prove -



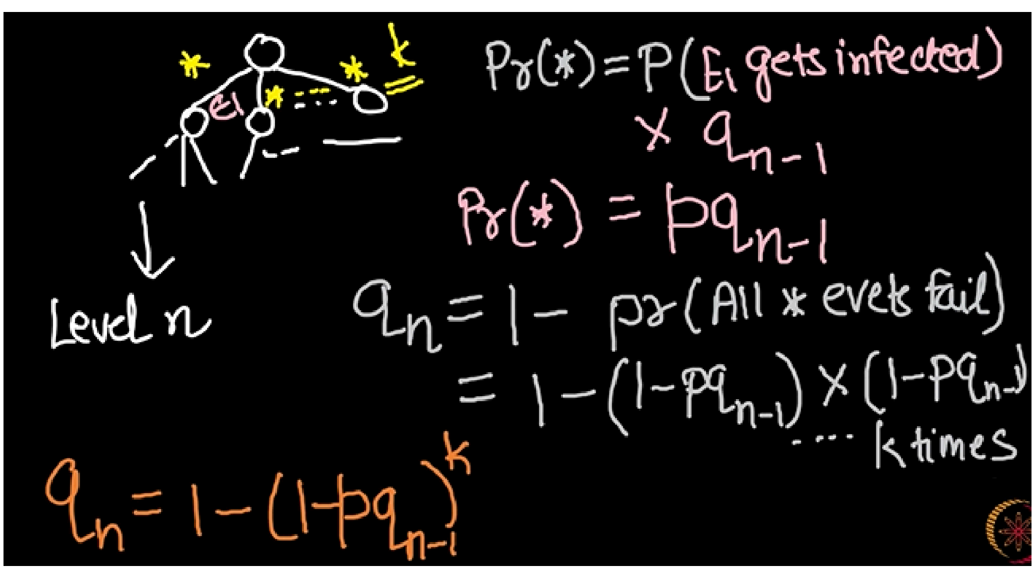


Define an event - * (star) event - an event that includes this edge in the path to infect at level n.
To infect at level n, there should exist a path





For P (from u …) = q) n-1



Lecture 140 - Analyzing basic reproductive number (3)

$R_0 < 1 \quad q^* = 0$

$R_0 > 1 \quad q^* > 0$

$$q^* = \lim_{n \to \infty} (q_n)$$

$$q_n = 1 - (1 - pq_{n-})^k$$

$$\rightarrow q_n = 1 - (1 - pq_{n-1})^k$$

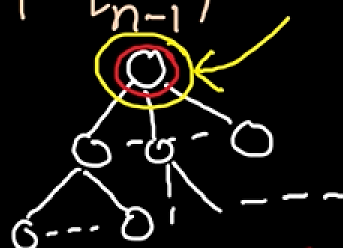
$q_0 = 1$

$q_1 = 1 - (1 - pq_0)^k$

$q_2 = 1 - (1 - pq_1)^k$

$\vdots$

q^k



$P \to n$ level

$q_0 = 1$

$q_1 = 1 - (1 - pq_0)^k$

$q_2 = 1 - (1 - pq_1)^k$

$\vdots$

$q_n = 1 - (1 - pq_{n-1})^k$

$$y = f(x) = 1 - (1 - px)^k$$

$q_1 = f(q_0)$

$q_2 = f(q_1) = f(f(q_0))$

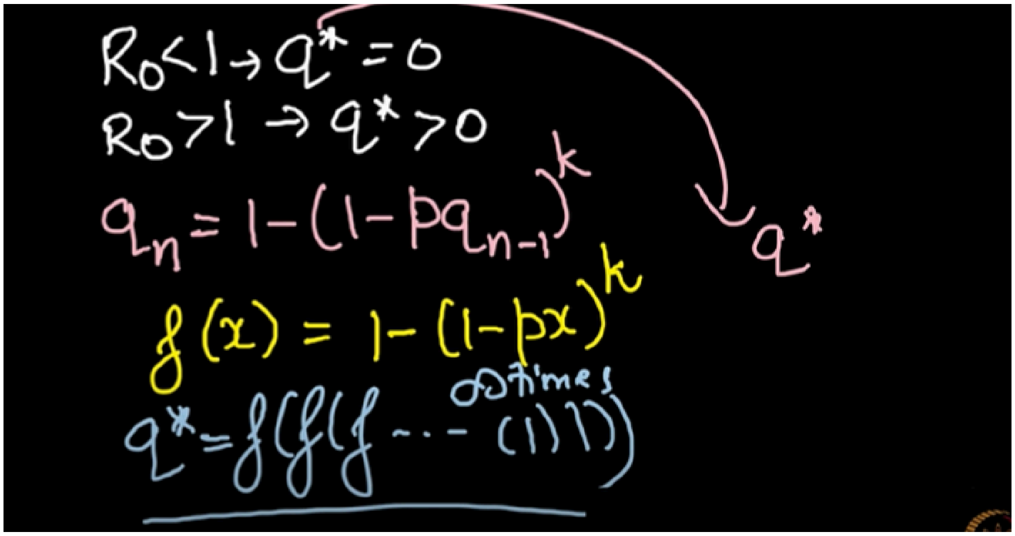
$\vdots$

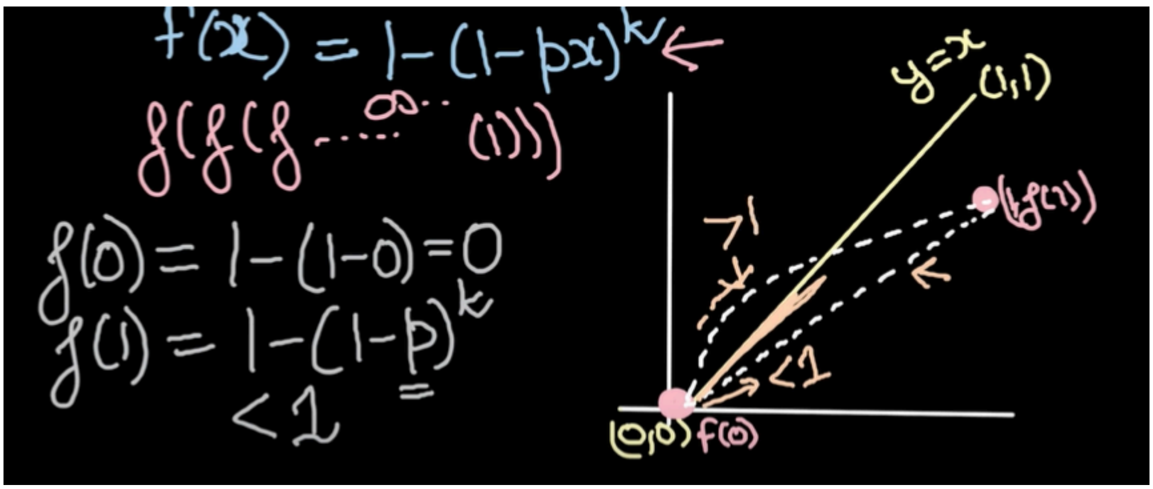
$q_3 = f'''(q_0)$

$q^* = f(f \cdots (q_0))$ ∞ times

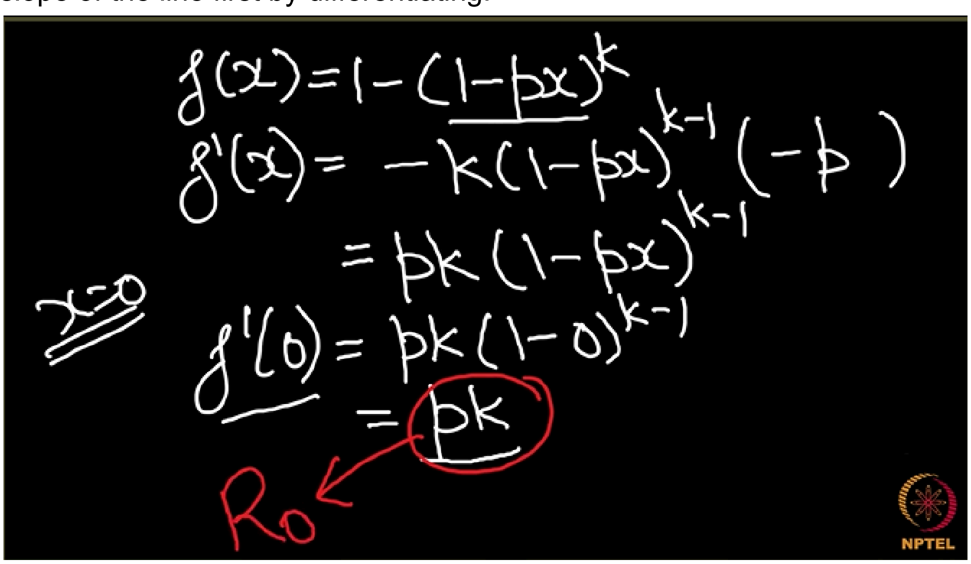
$q^* = f(f(f \cdots (1)))$ ∞ times

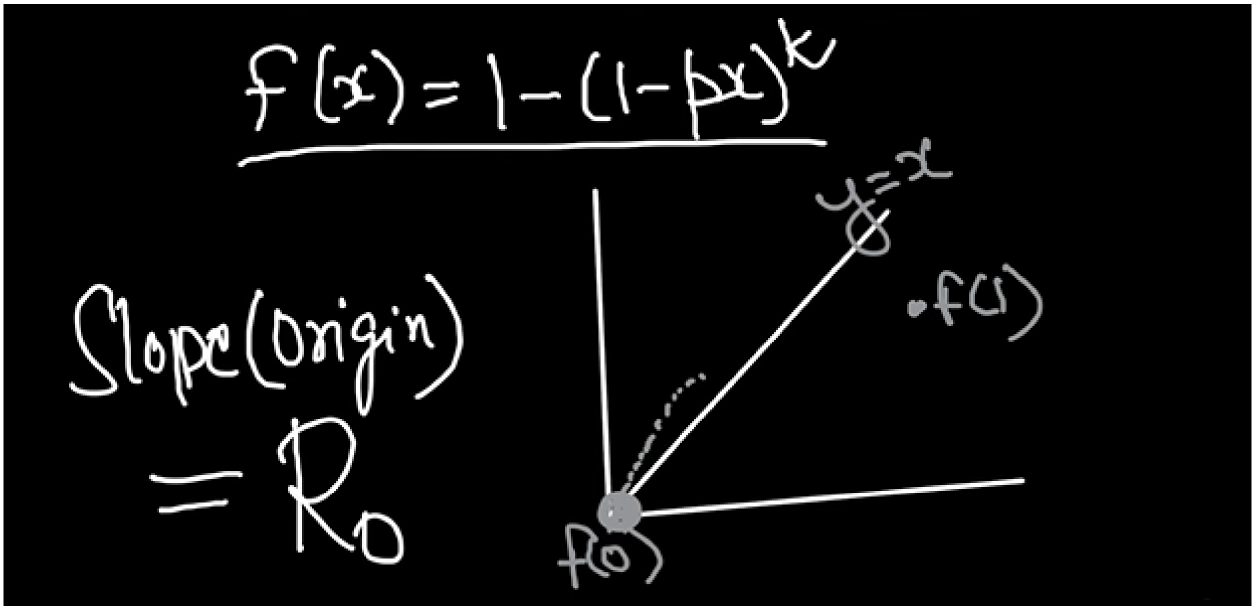
Lecture 141 - Analyzing basic reproductive number (4)



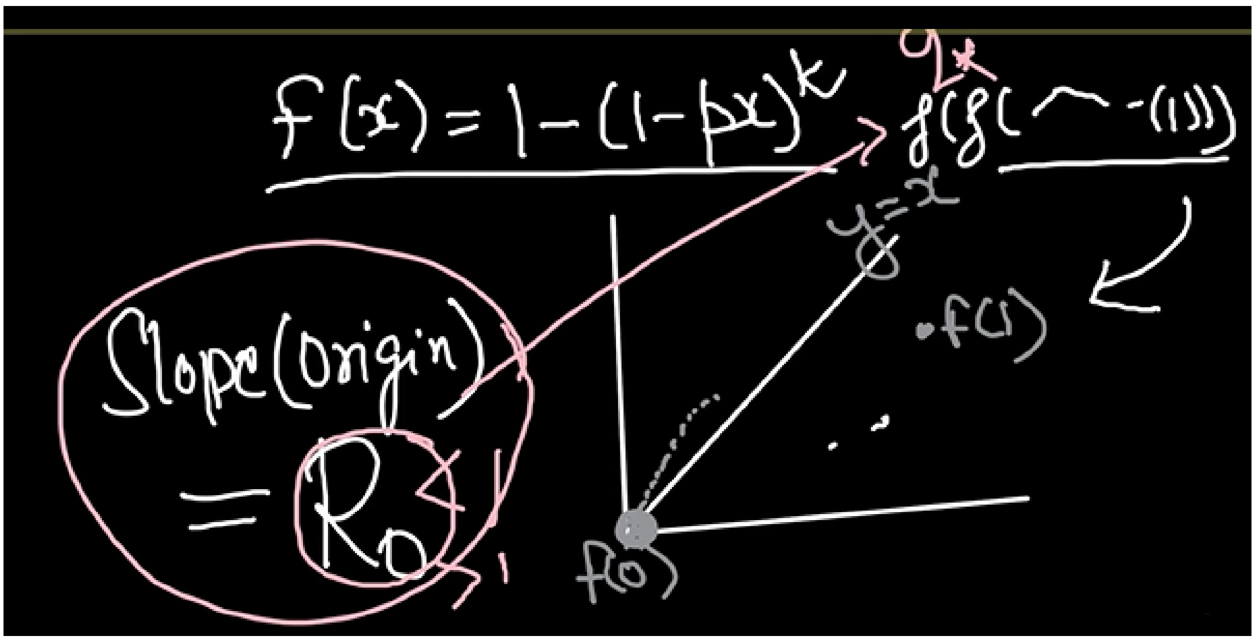


The graph drawn can be straight line(if slope <1) or parabola (if slope>1). Basically find the slope of the line first by differentiating.



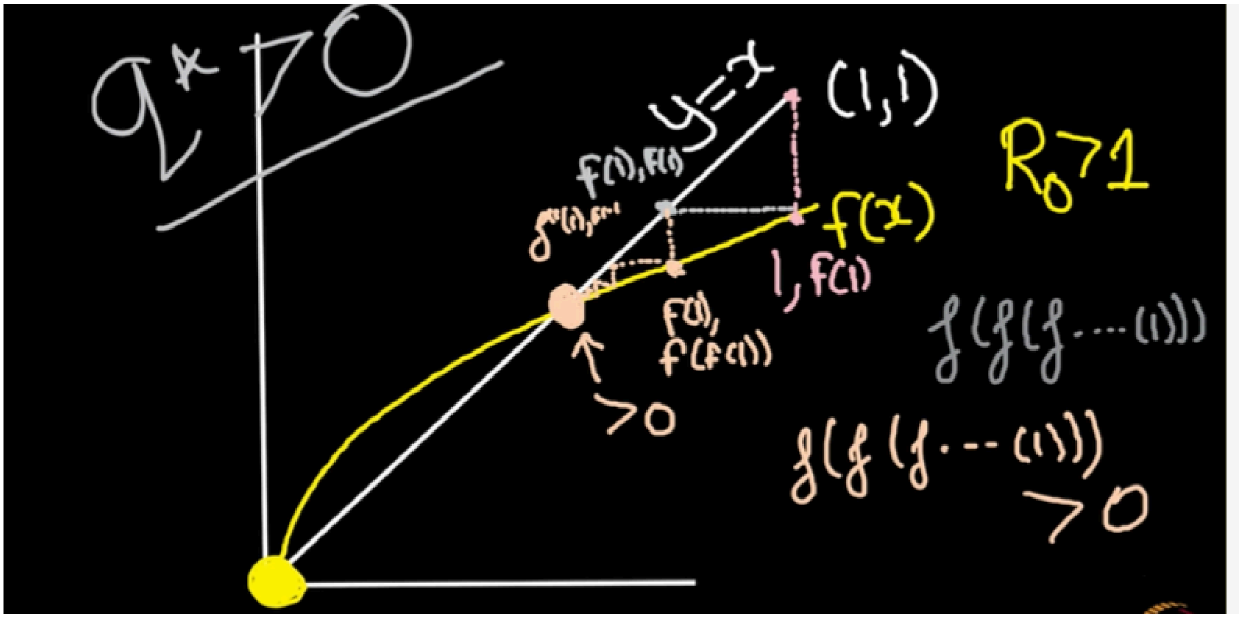


$f(x) = 1 - (1 - px)^k$
$y = x$
$f(1)$
Slope(origin)
$= R_0$
$f(0)$

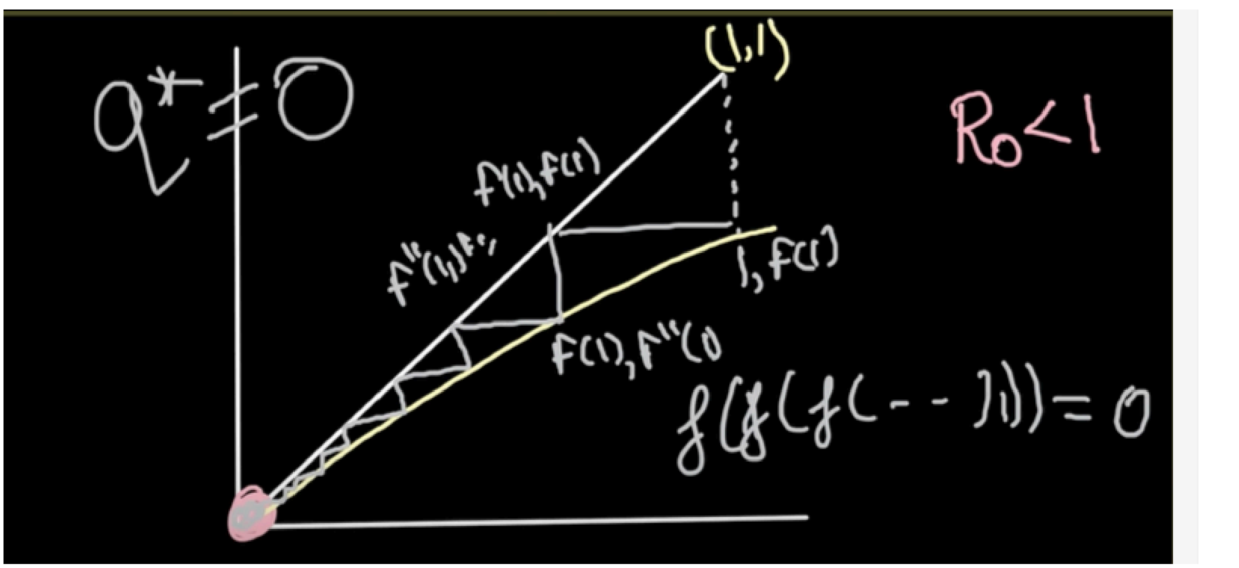


$f(x) = 1 - (1 - px)^k$
$y = x$
$f(1)$
Slope(origin)
$= R_0 > 1$
$f(0)$

Lecture 142 - Analyzing basic reproductive number (5)
WHEN R0 > 1



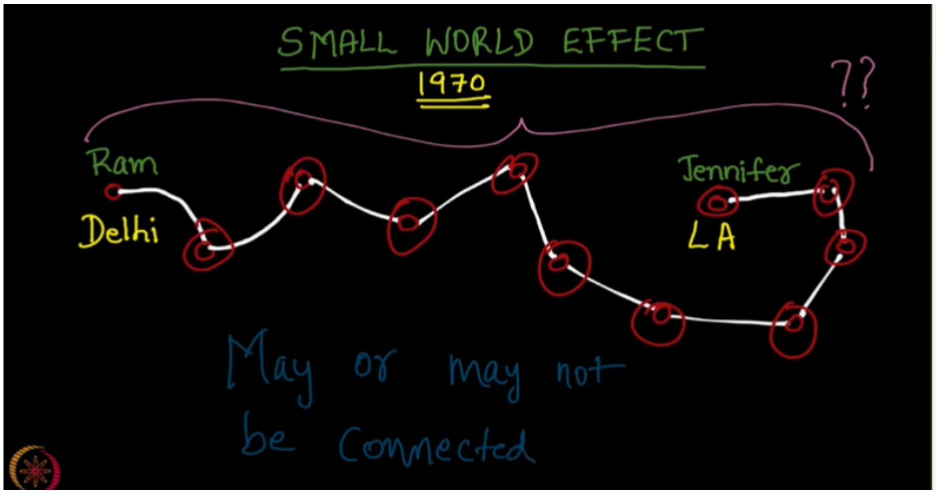
R0 <1



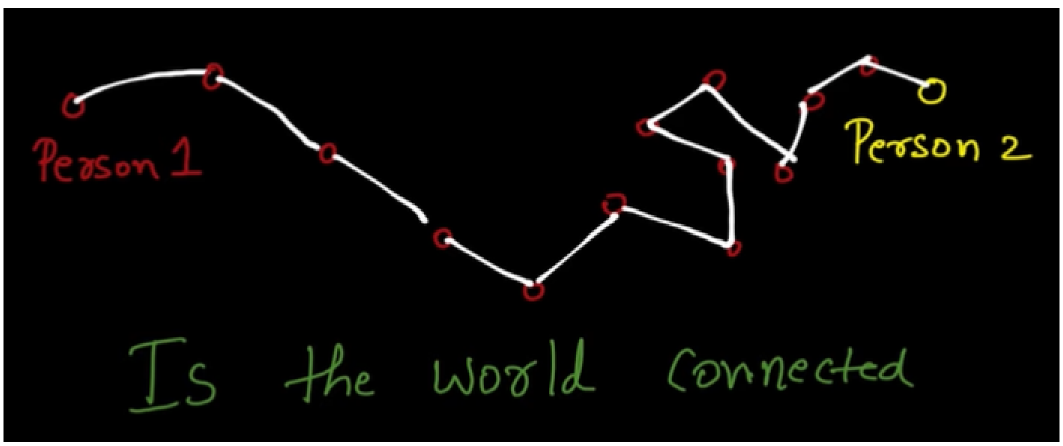
WEEK 11 - SMALL WORLD EFFECT

Lecture 143 - Introduction

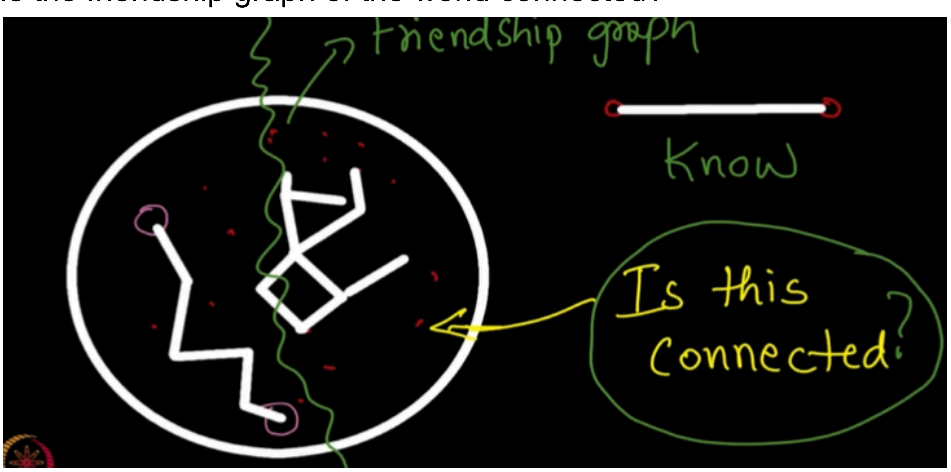
U in india, want to connect jennifer from LA in 1970s and connect her through people knowing each other in a chain



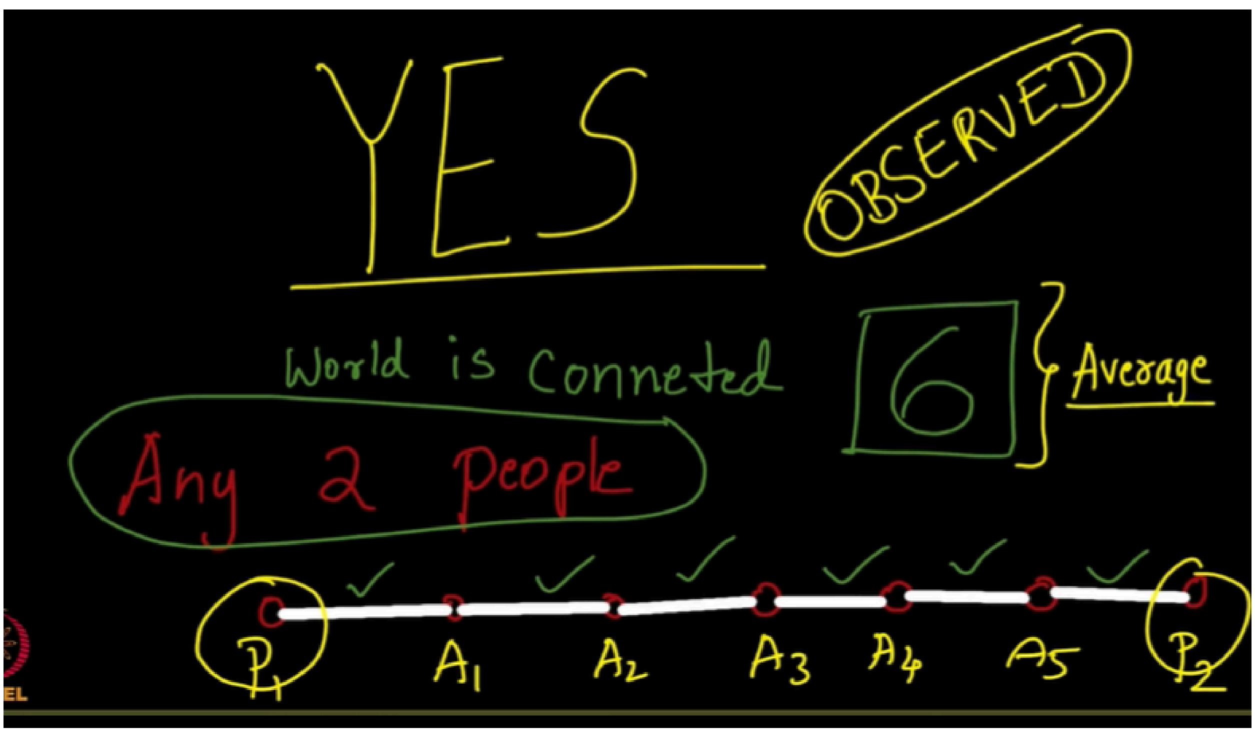
Is the world connected? Take 2 random people from different areas in the world.



Is the friendship graph of the world connected?

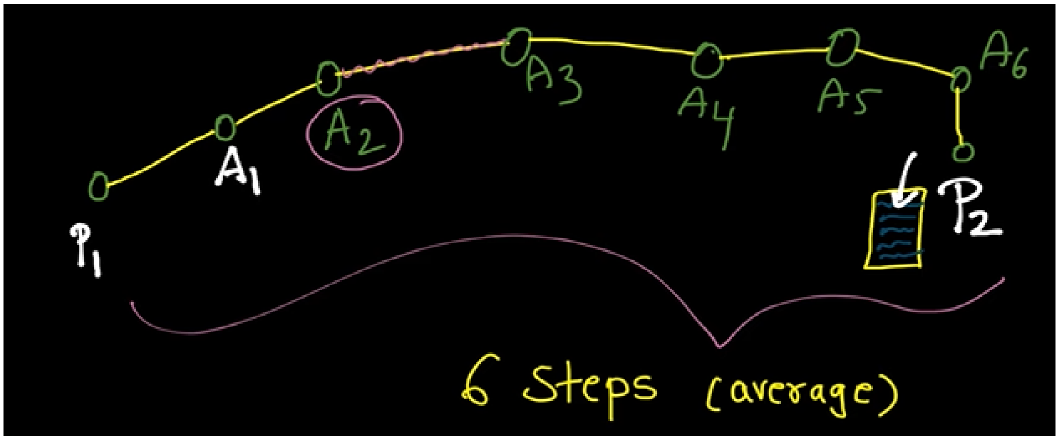


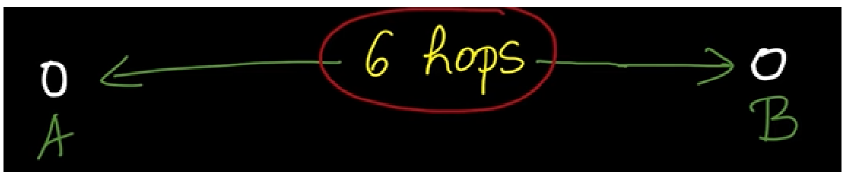
Is this always a small number of people = 4?
Scientists observed on an avg. 6 people in between can connect any random 2 people from the world .

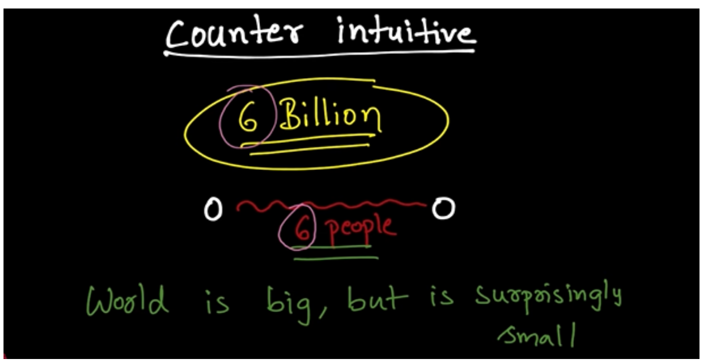


Lecture 144 : Milgram's Experiment

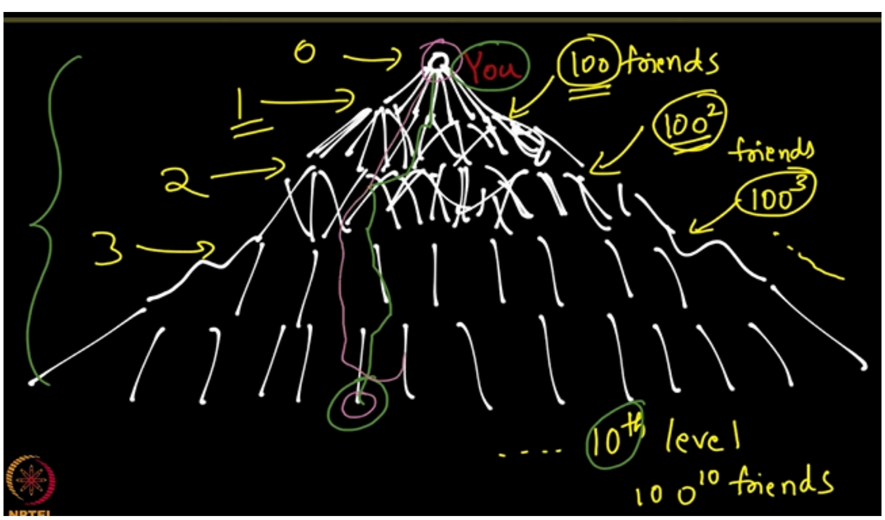
Expt - p1 and p2 dont know each other, and p1 has to send a leter to her. He has to pass on to someone p1 might know, and then keep on passing it





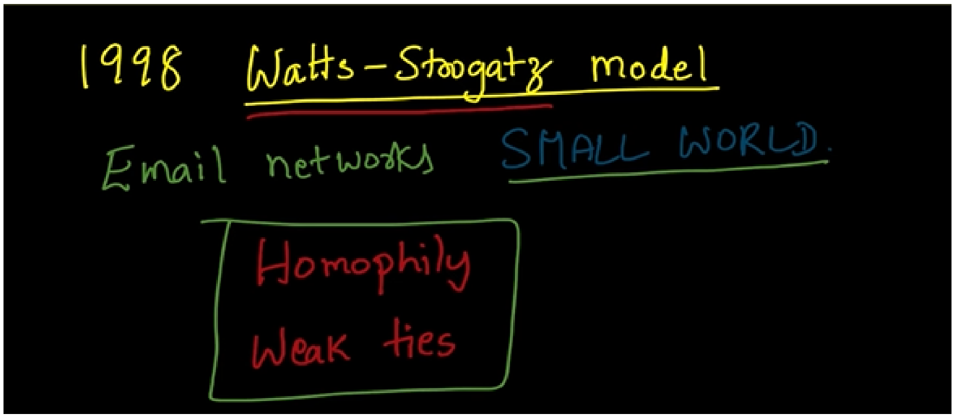


sounds counterintuitive but is obvious

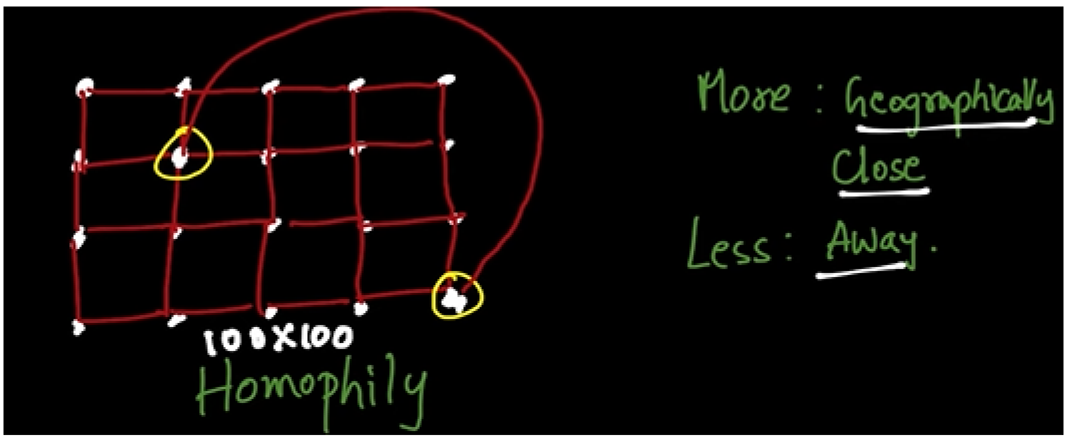


thus, not counter intuitive

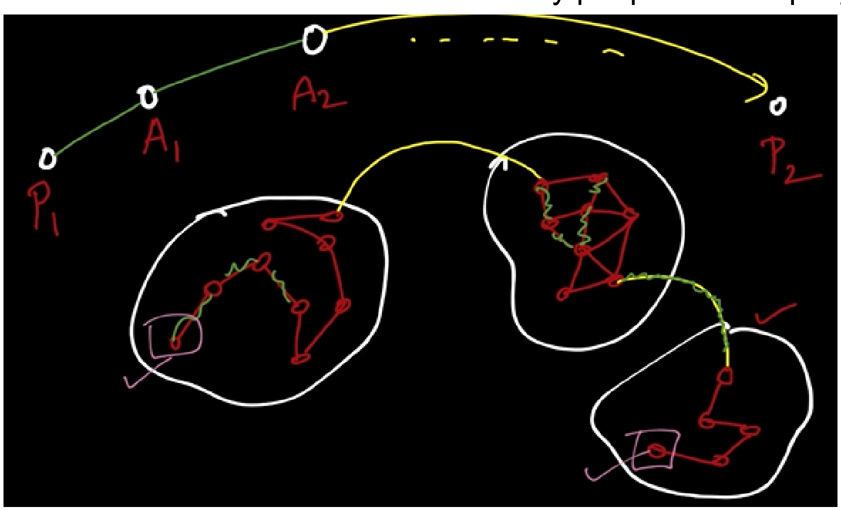
Lecture 145 : The Reason
Expt was redone in 1998 on email ntwrks



these 2 concepts were the reason of small world phenomenon acc to them

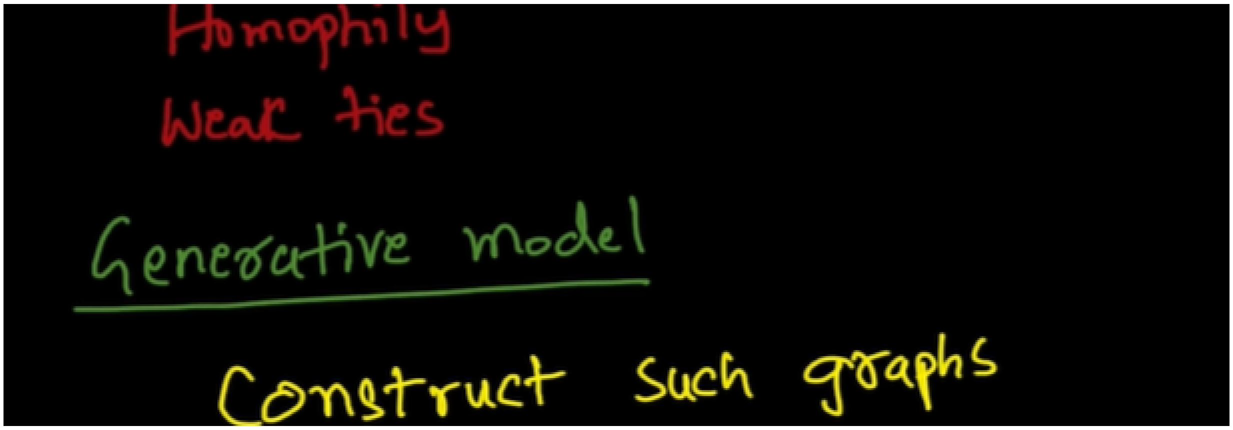


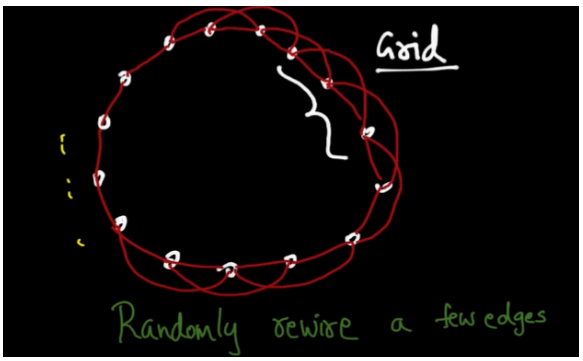
You can also have friends from far away people = homophily (reason)



Nodes exhibit homophily and this combined with weak ties result in small world phenomemn

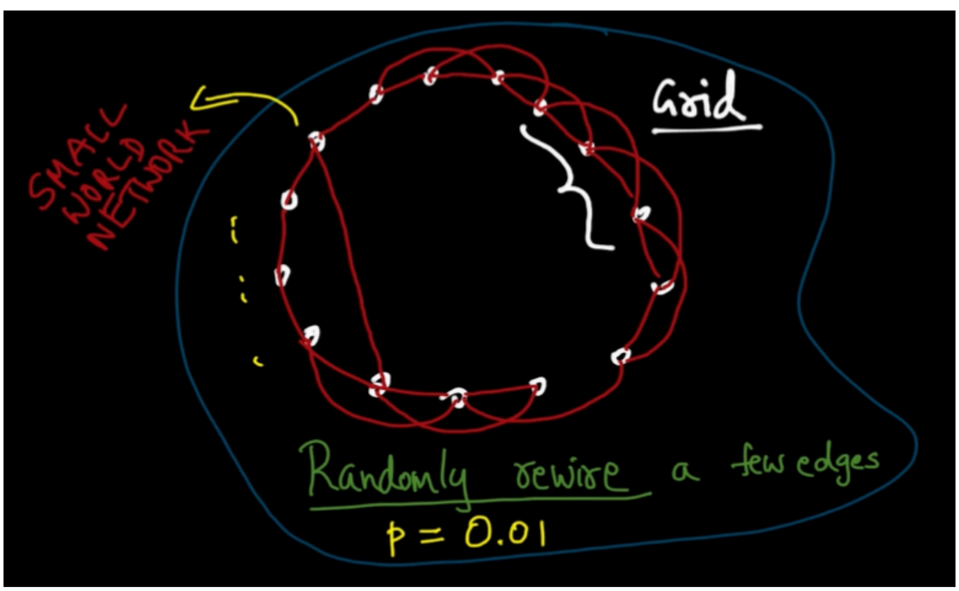
Lecture 146: The Generative Model = algo by which u can construct such small world graphs



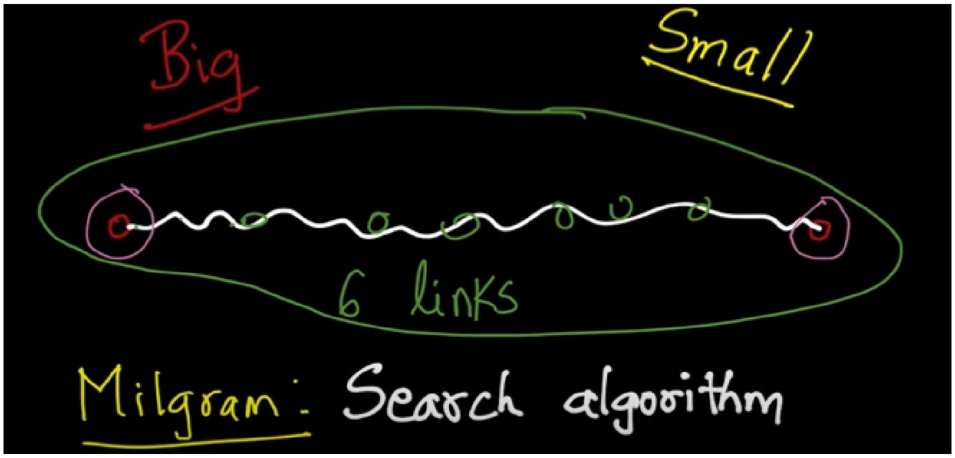


let p =0.01 for rewiring

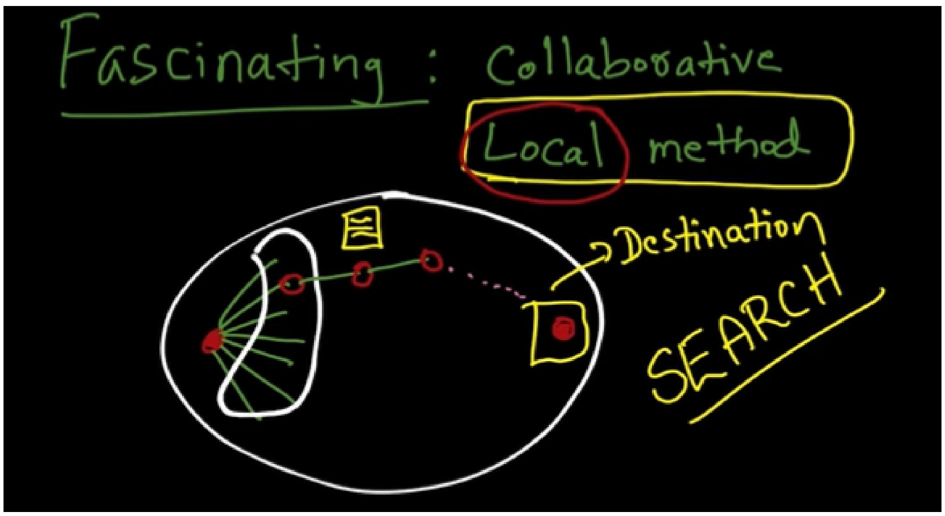
Whenever rewiring was done it always resulted in small world n/w

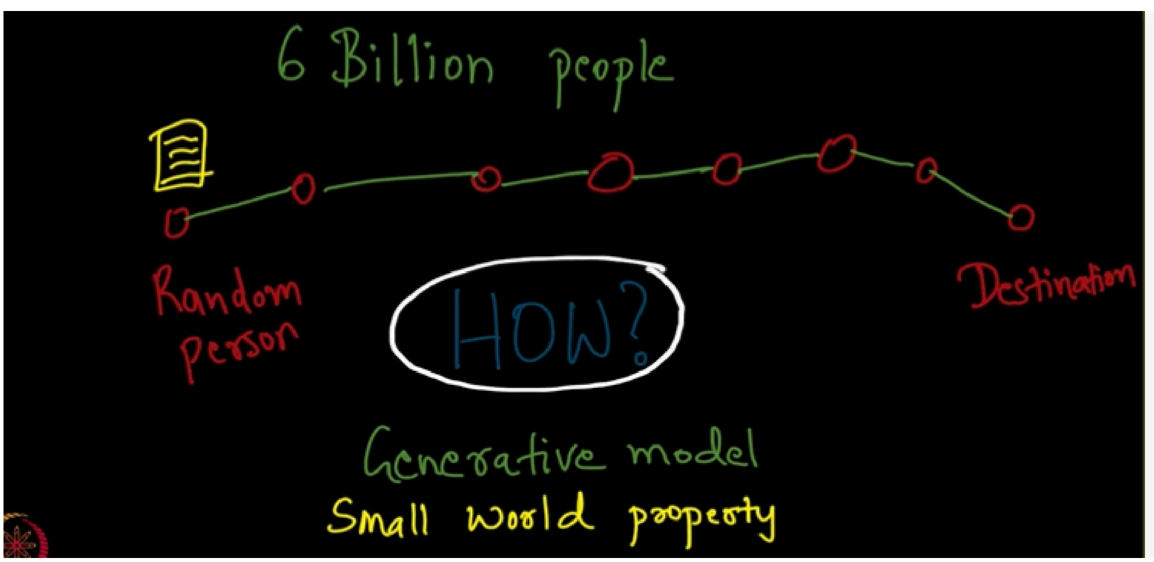


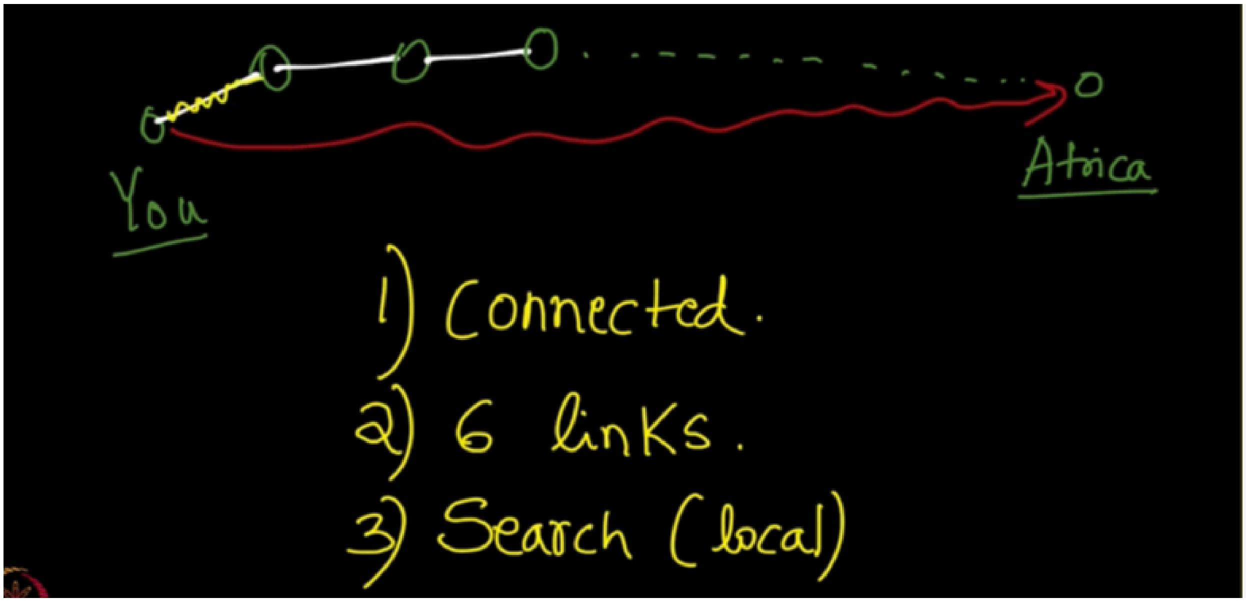
Lecture 147 : Decentralized Search - I



It is a search algo, where one searches the frnd to send further and its completely local
It is a local method - no one knows the whole graph, he only knoes his own friends

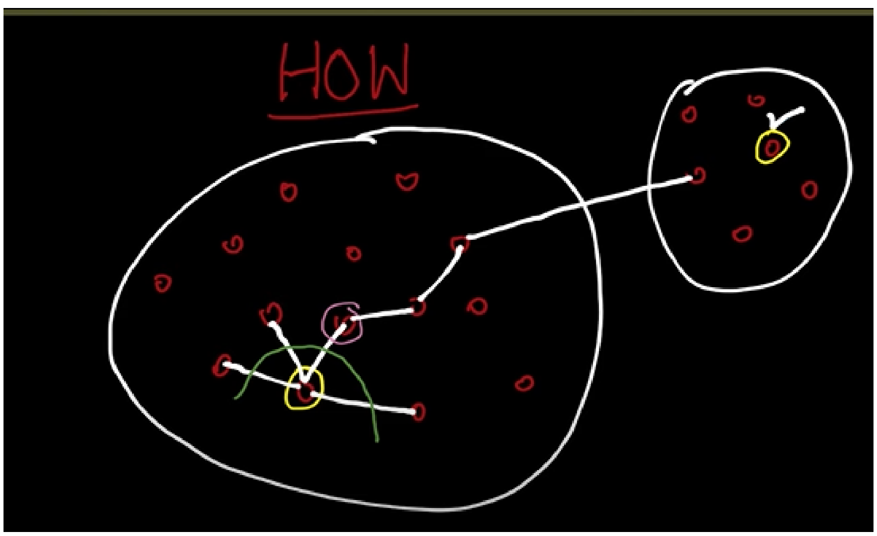


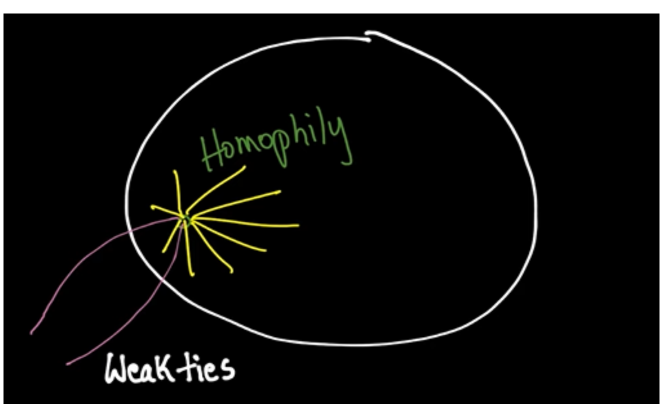


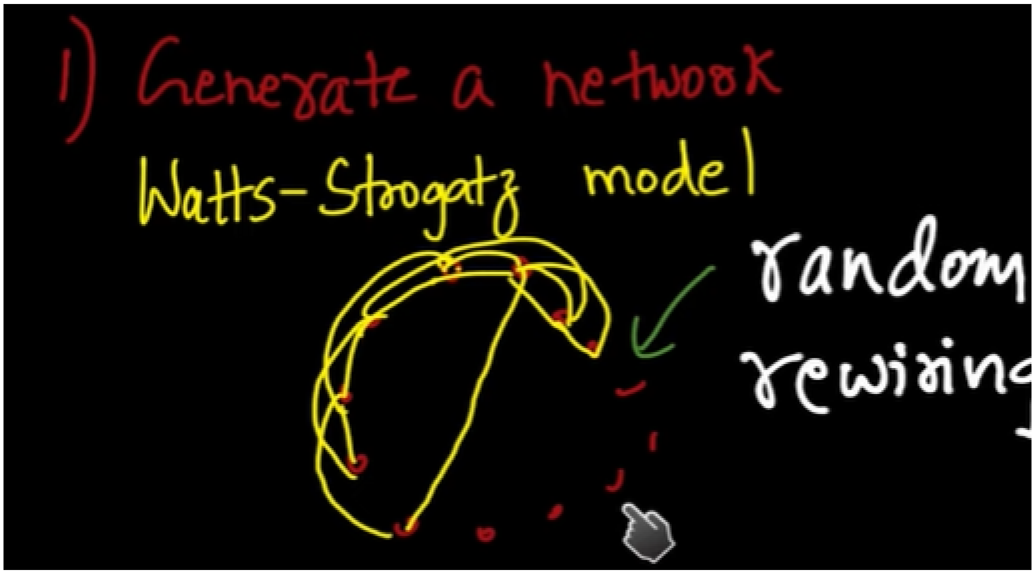


Lecture 148 : Decentralized Search - II

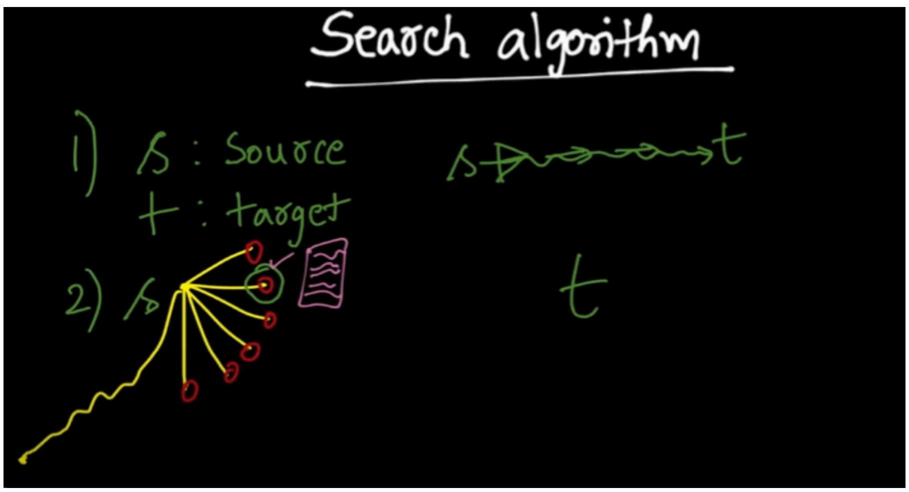
HOW ?? - A person , purposely chooses that person to forward who can / may send it further to the different community







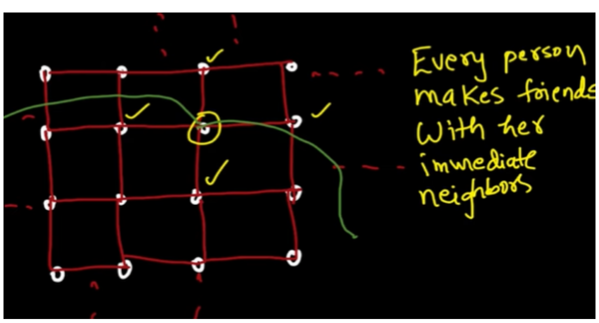
Random rewiring is the fact u have connections outside to a far off place

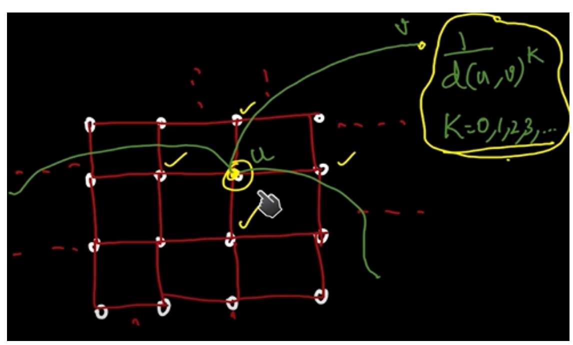


Pass letter to someone who is close to t.

But there is a possibility that u keep on going far far away from t - but this actually never happens in real life.

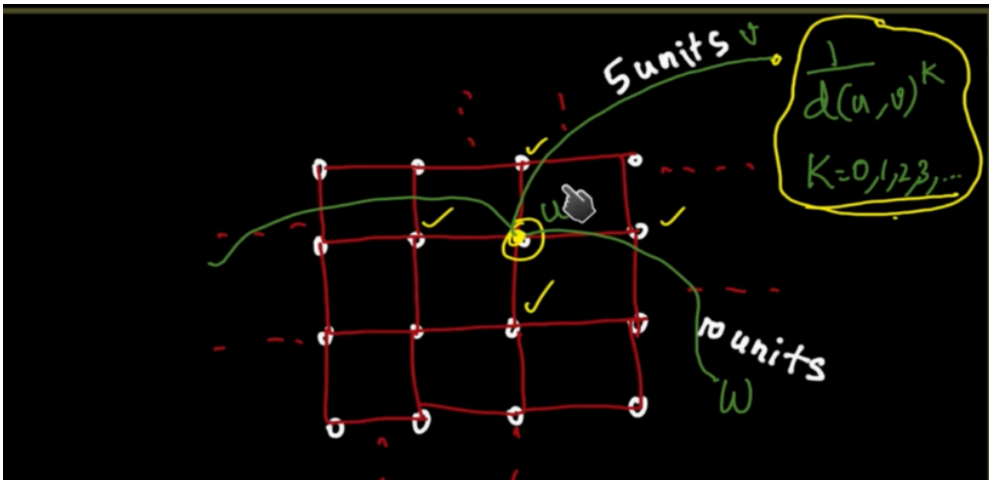
Green line = connection with far away node



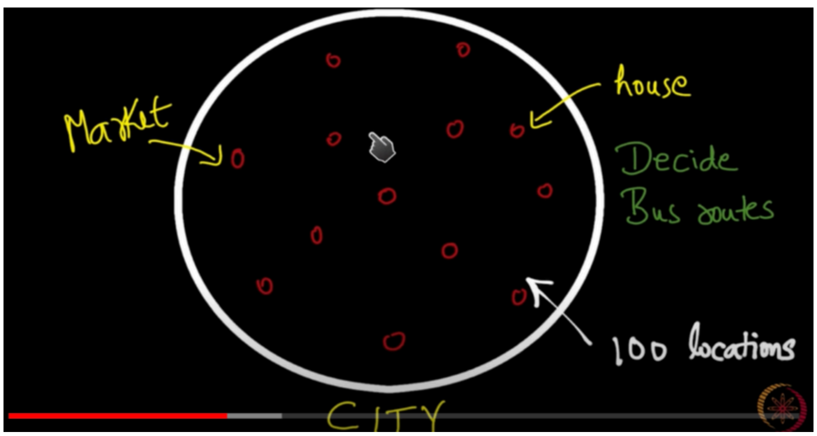


Put an edge to the far away node with prob = 1 / distance (u,v) ^ k

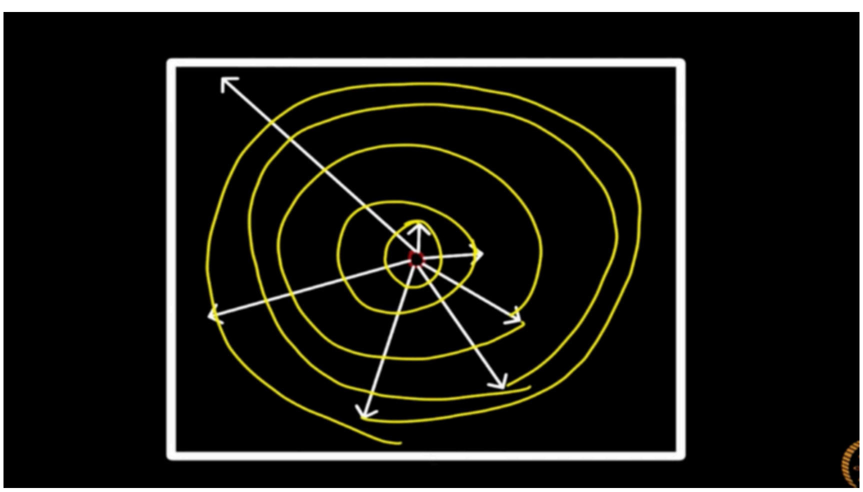
Lecture 149 : Decentralized Search - III
Acc to the above formulae, far is the node, lesser is the prob to become friends with.



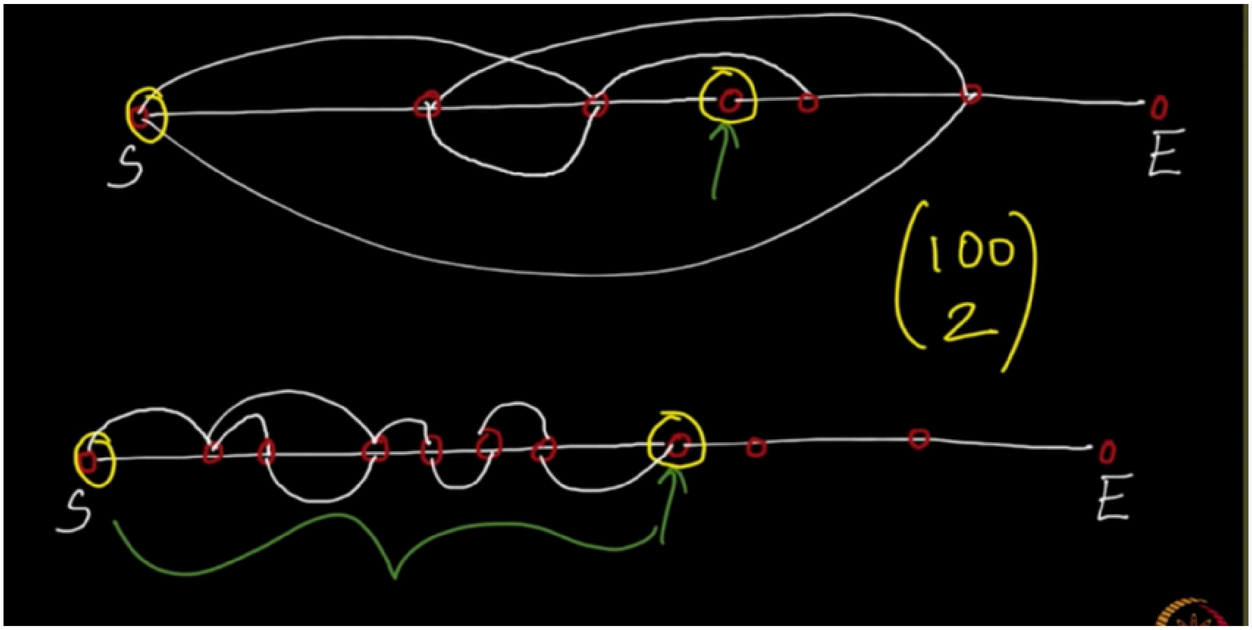
Prob to be frnd with v > with w
If u want to decide the bus routes of a new city with 100locations



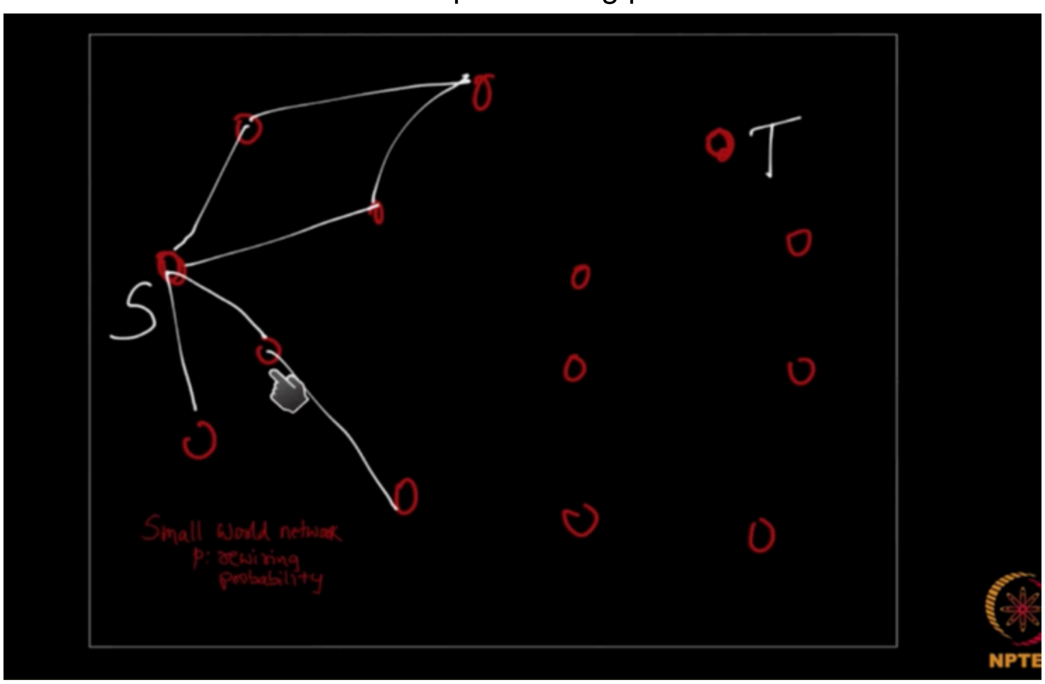
You do it such that there bus on all the concentric circles covering all areas and change the buses



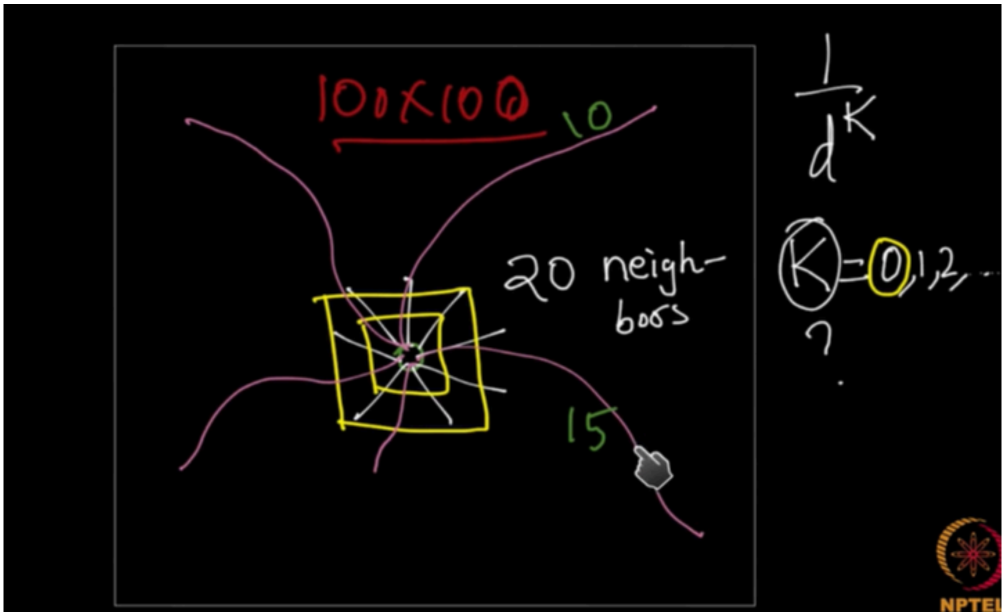
Assume metro system in the city - u want to reach green



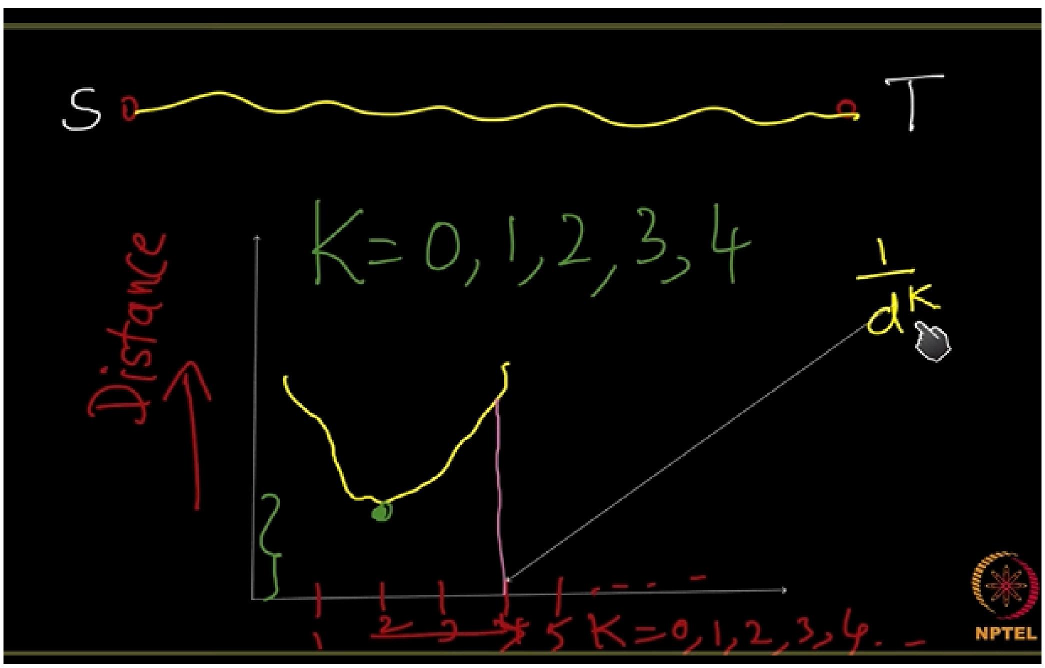
For 1st, you go ahead and have to come back
For 2nd, you have to make many jumps/ changes which is very tedious. If 100 locations then 100C2 wil be needed
Best way to put metro in city -
Assume a small world n/w with p = rewiring prob



Closer the vertices more are the edges



what is ideal k?
If k is very large, prob reduces a lot and nodes become very improbable
On plotting the graph, when k =2, dist(path length from S to T) is minimum. Thus k =2 is the best value
For 2D - k=2 ;; for 1D , k=1 (best values)



Assignment Q7 - In a 1-D Watts-Strogatz model, each node is connected to k neighbors on each side in a ring topology. Therefore, each node contributes k edges, and since there are n nodes, the total number of edges in the network is 2n * k.

So, the network will contain 2n edges where k can be any positive integer.

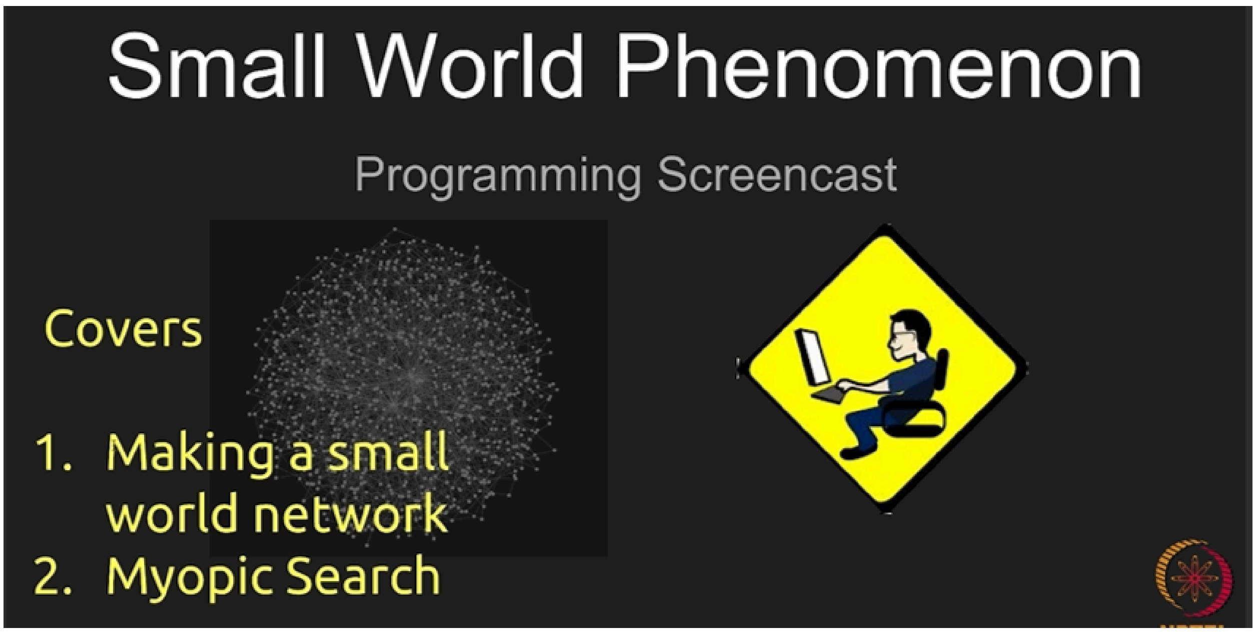
A **small-world network** is a graph characterized by a high clustering coefficient and low distances. On an example of social network, high clustering implies the high probability that two friends of one person are friends themselves. The low distances, on the other hand, mean that there is a short chain of social connections between any two people (this effect is known as six degrees of separation).[1] Specifically, a small-world network is defined to be a network where the typical distance L between two randomly chosen nodes (the number of steps required) grows proportionally to the logarithm of the number of nodes N in the network, that is:[2]

$$L \propto \log N$$

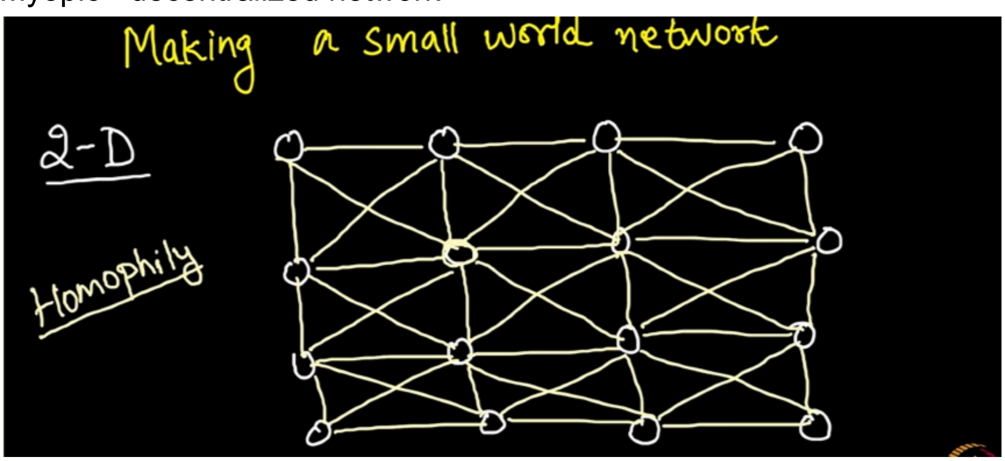
while the global clustering coefficient is not small.

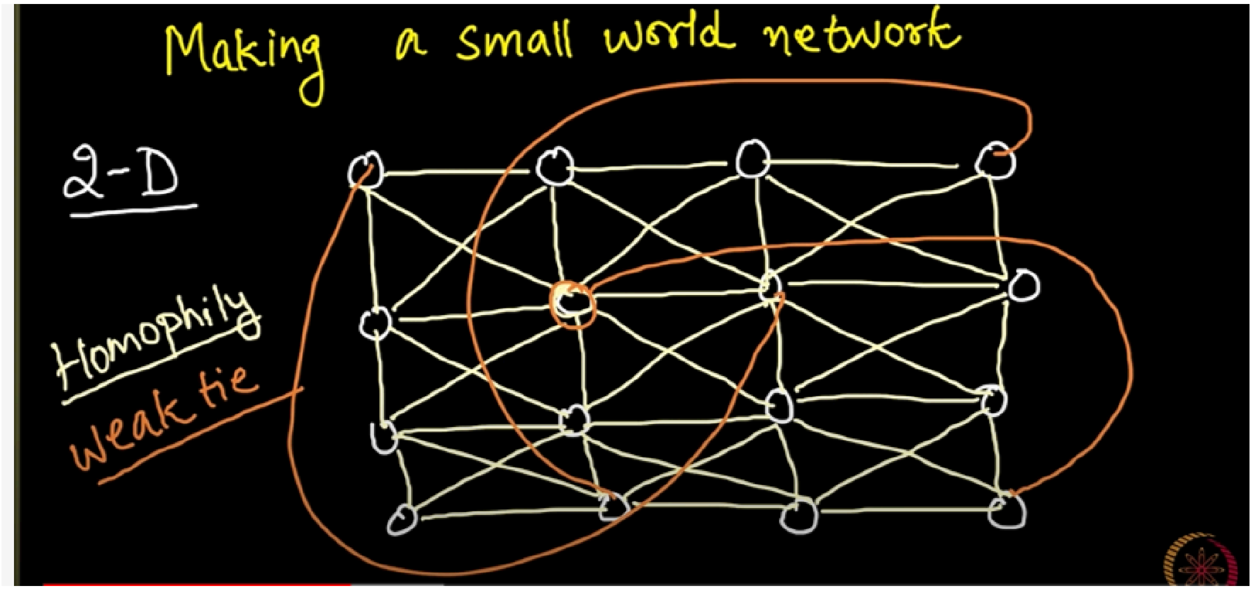
WEEK 12

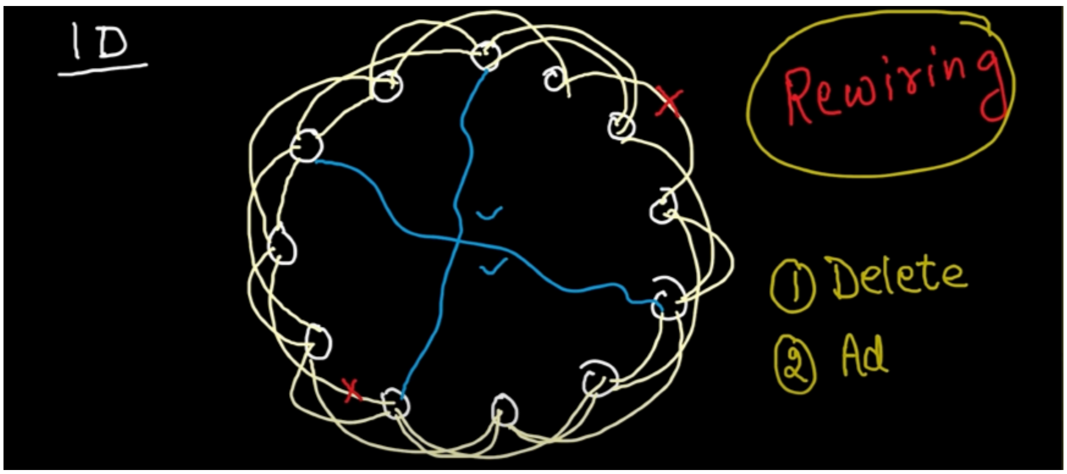
Lecture 150 : Programming illustration- Small world networks : Introduction

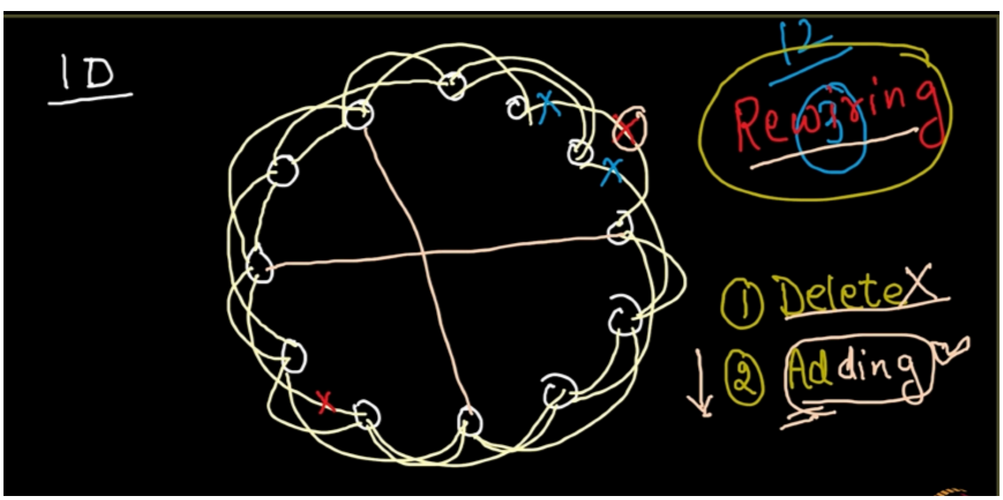


Myopic - decentralized network

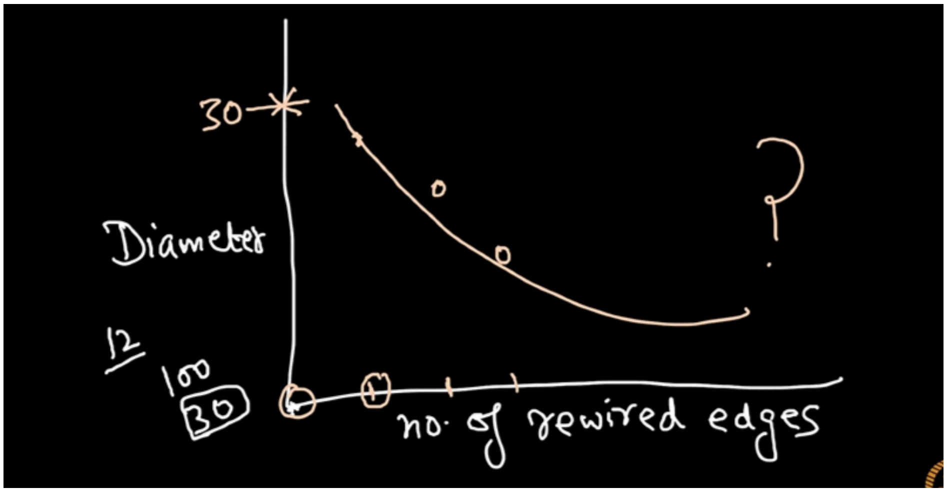




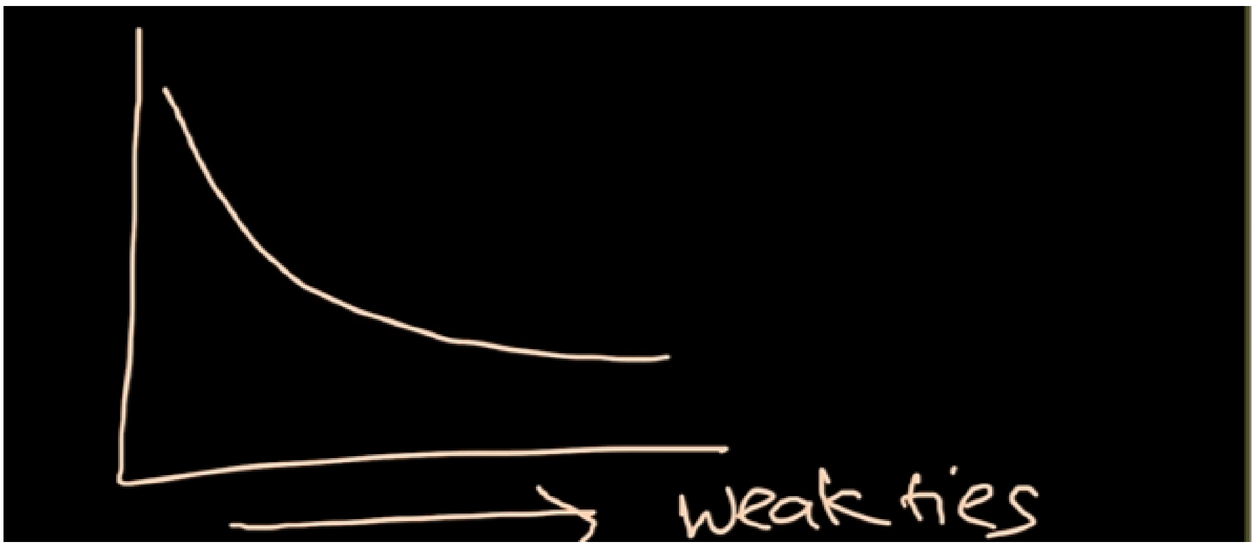




We can also add random edges rathe rthan rewiring = this helps in reducing the diameter.

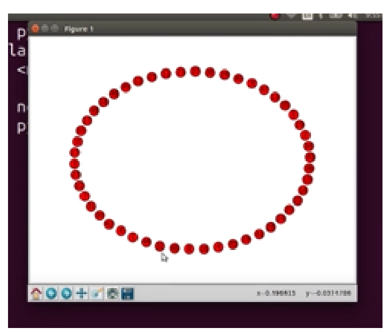


Diameter vs no. of weak ties



Lecture 151 : Base code

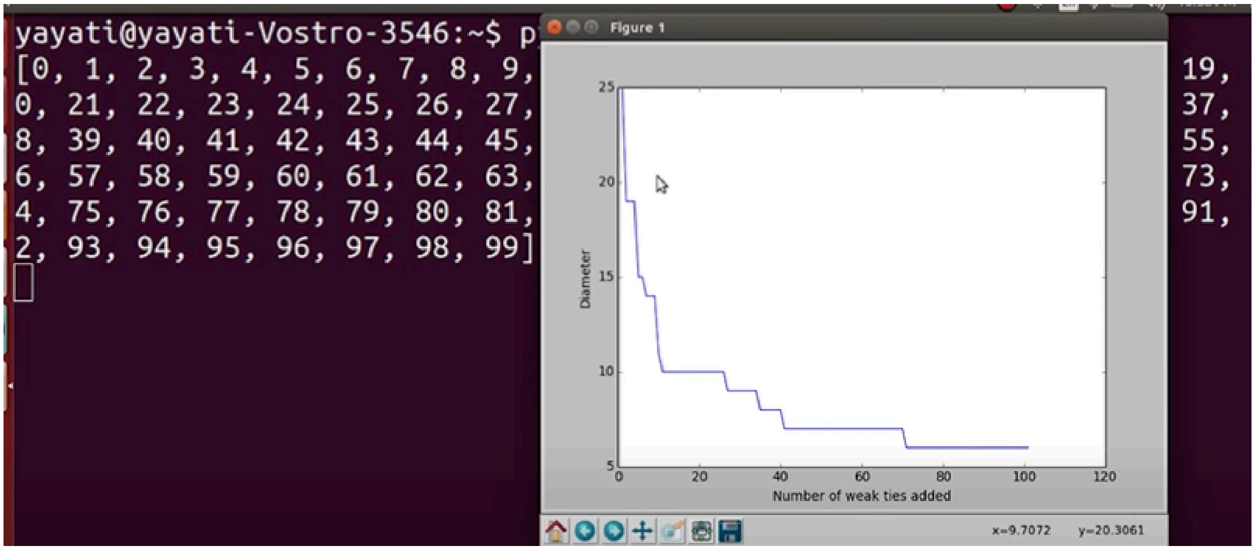
```
import networkx as nx
import matplotlib.pyplot as plt
G=nx.Graph()
G.add_nodes_from(range(0,50))
nx.draw(G)
plt.show()
```



Lecture 152 : Making homophily based edges
Adding edge acc to homophily, 2 nodes towards right and 2 towards left

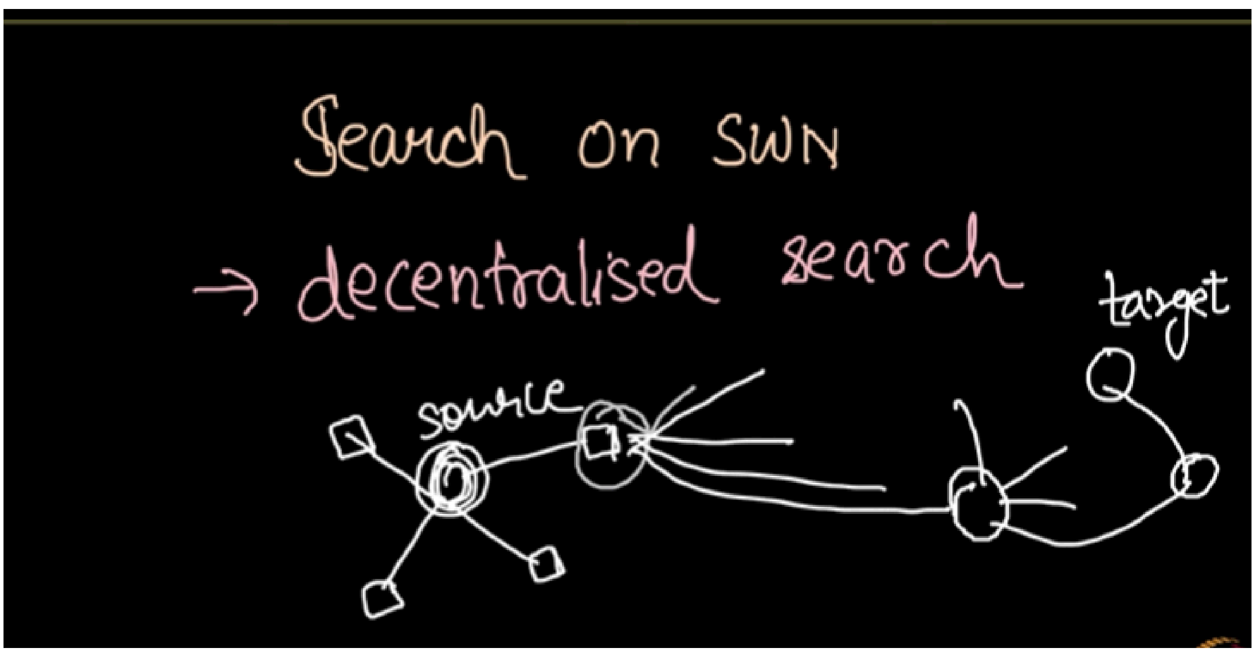
Lecture 153 : Adding weak ties

Lecture 154 : Plotting change in diameter



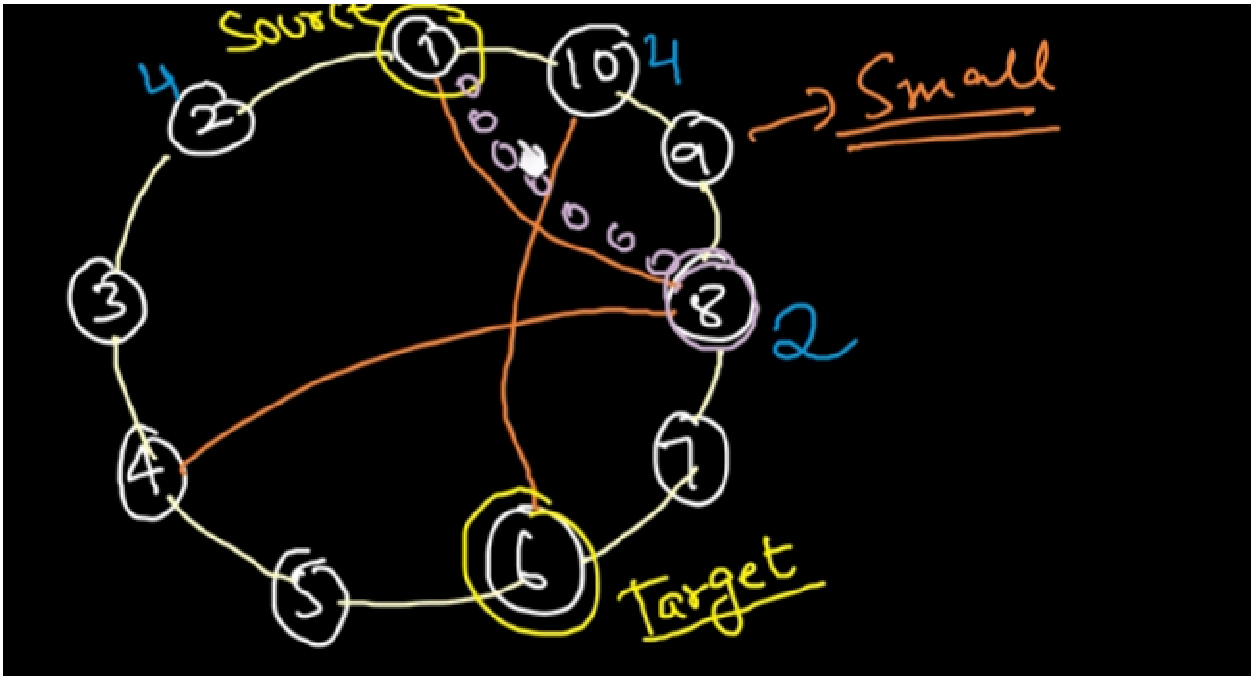
Initially it was diameter of 25 . and then it reduces quickly

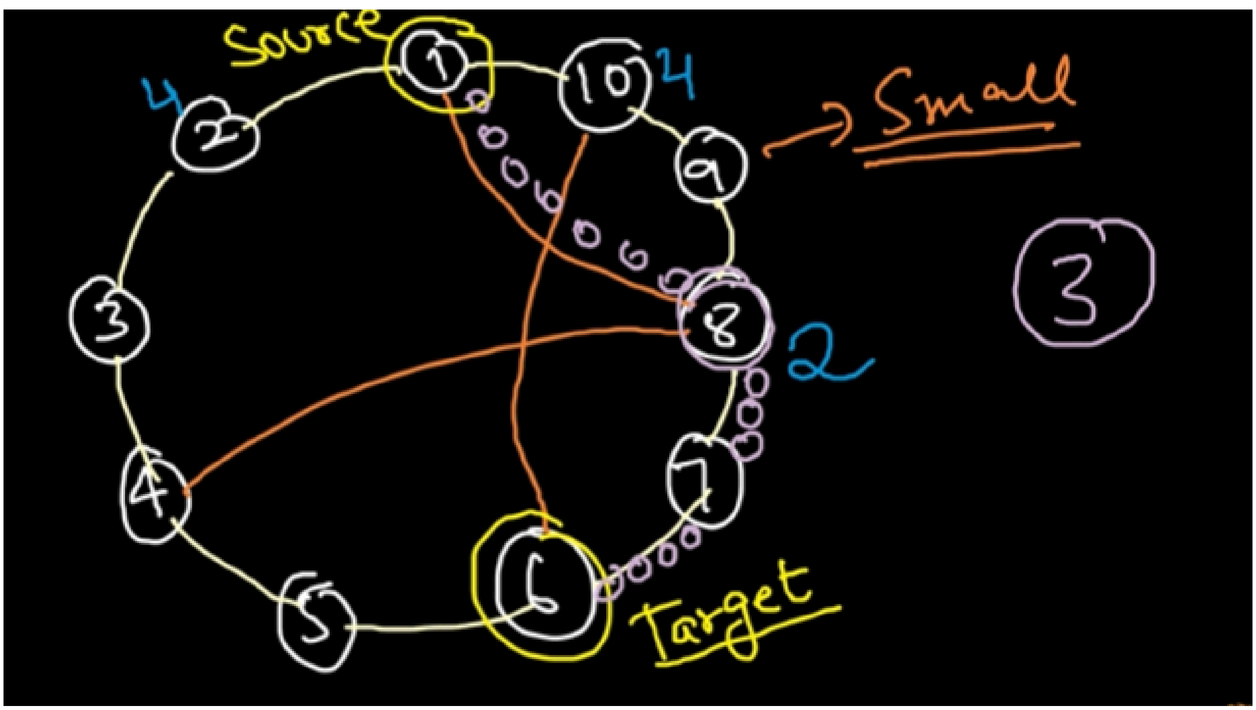
Lecture 155 : Programming illustration- Myopic Search : Introduction



Doing myopic = decentralized search

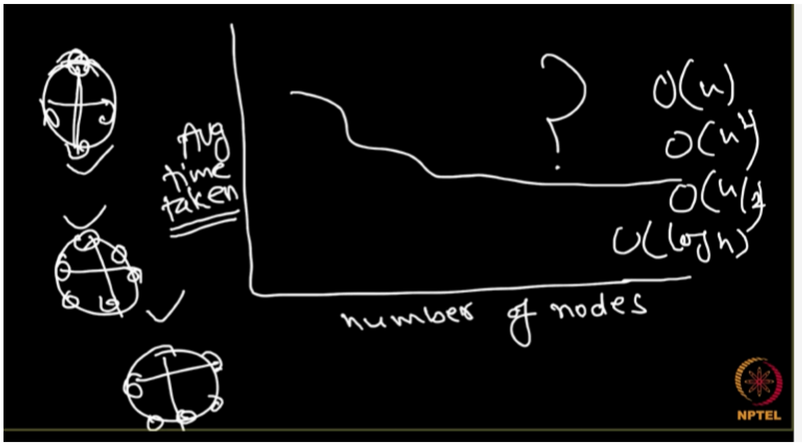
Node has knowledge about its neighbours distance from the target
Dist from 2 to 6 - 4
Dist from 10 = 4
Dist from 8 = 2



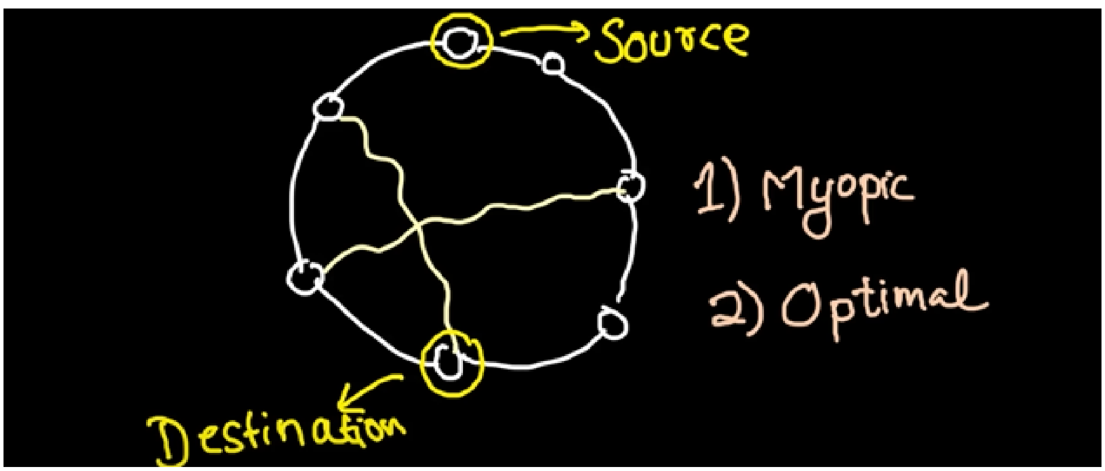


Length of path = 3

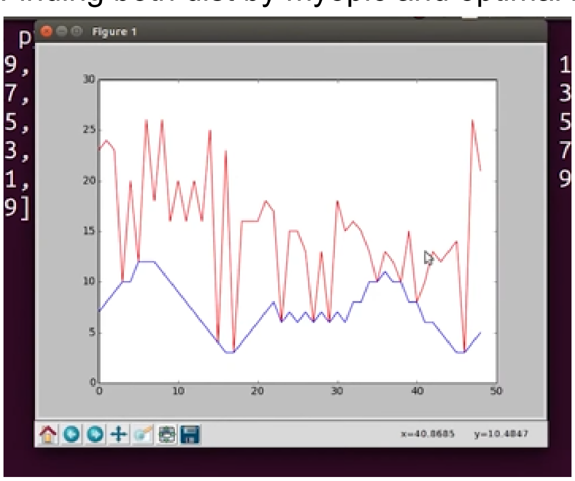
But this is not optimal. You can have the optimal pathe from 1-10-6. But this can be done only if someone ahs complete knowledge of neighbour and ntwrk



Lecture 156 : Myopic Search
Lecture 157 : Myopic Search comparision to optimal search

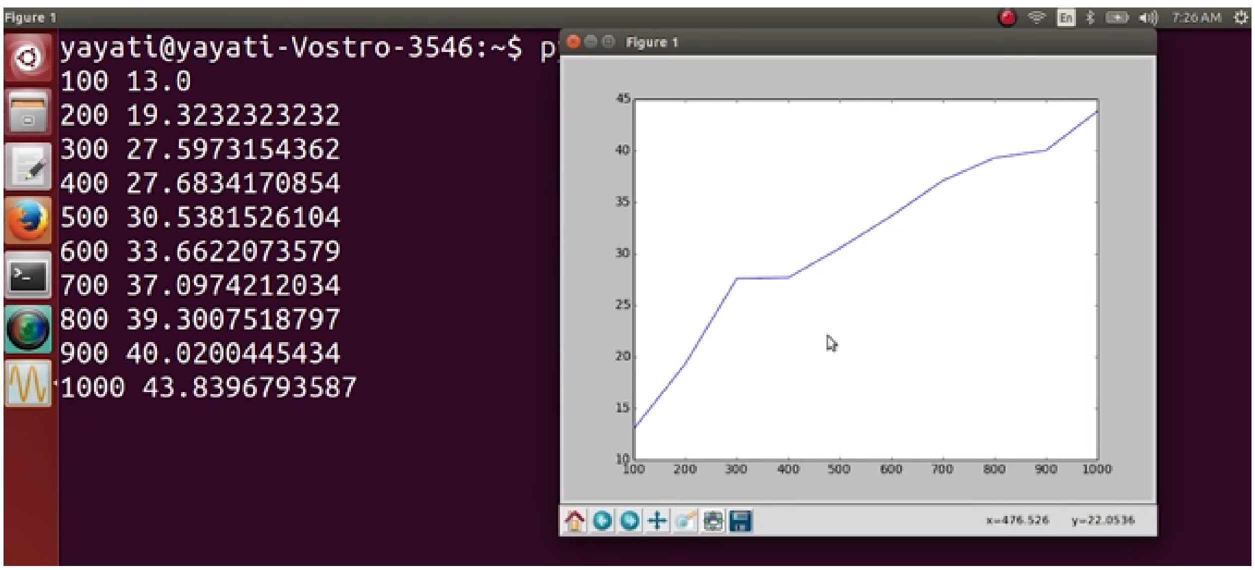


We taking 1 -100 notes, 10 weak te=ies. Taking diagonal 51 pairs - 0,49 ; 1,50; ….50,99
Finding both dist by myopic and optimal length and plotting

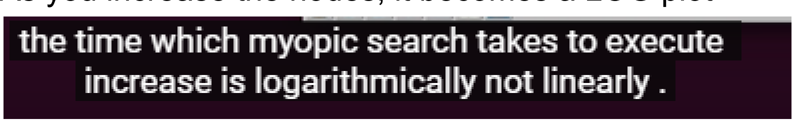


red = myopic ; blue = optimal

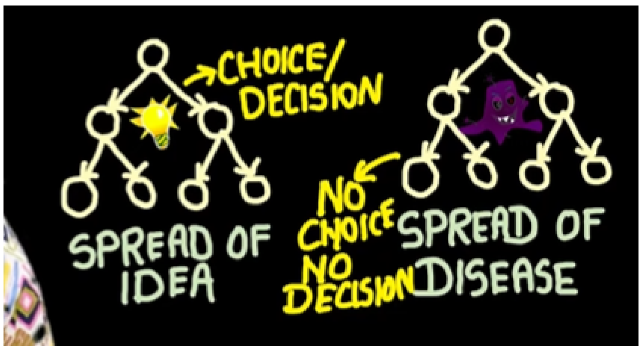
Lecture 158 : Time Taken by Myopic Search
Calculating avg time for diametrically opp points by myopic for 100, 200, 300 nodes,...
With 10,20,30… weak ties



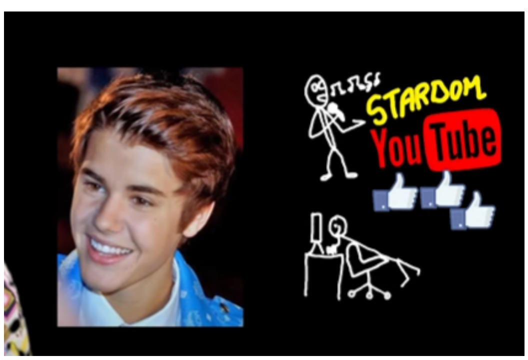
As you increase the nodes, it becomes a LOG plot



Lecture 159 : PseudoCores : Introduction
We studied -

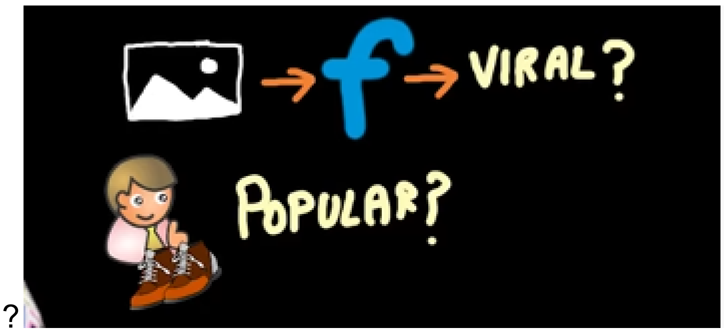


—-------------

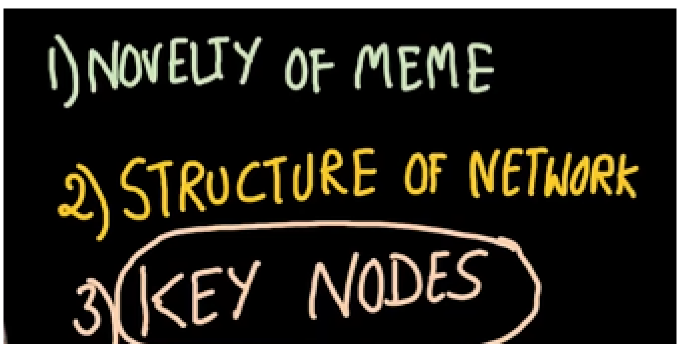
Justin bieber story 

Lecture 160 : How to be Viral

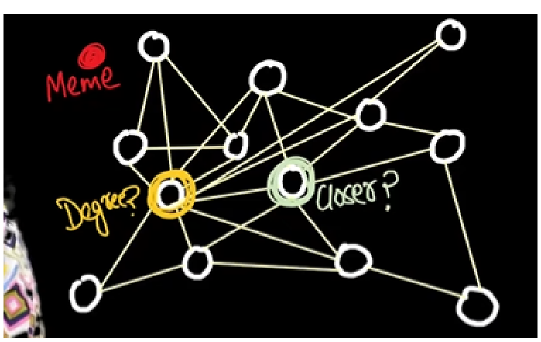
How to know our idea goes viral ?

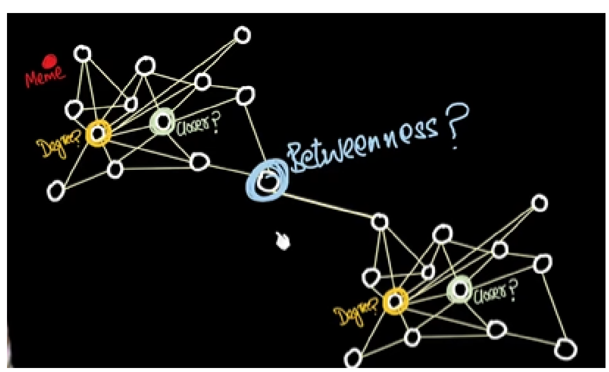


3 FACTORS - (quality of meme, structuore of ntwrk, key nodes)

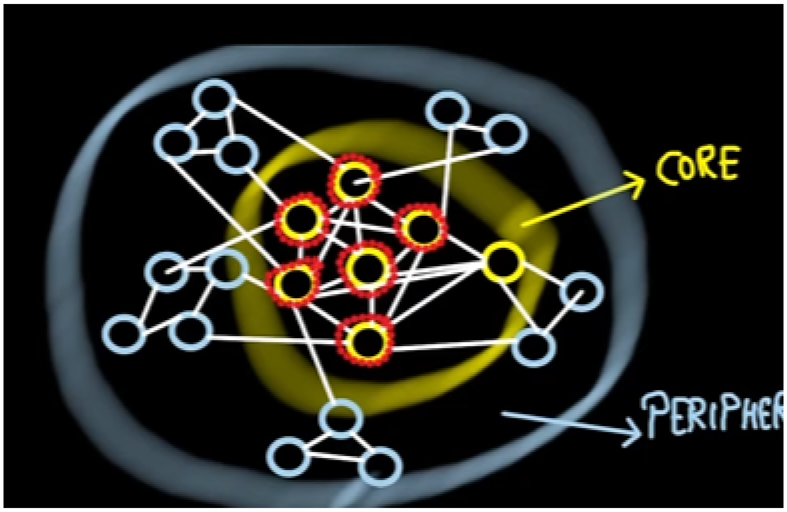


Lecture 161 : Who are the right key nodes?
Degree ? closer? High betweeneness?



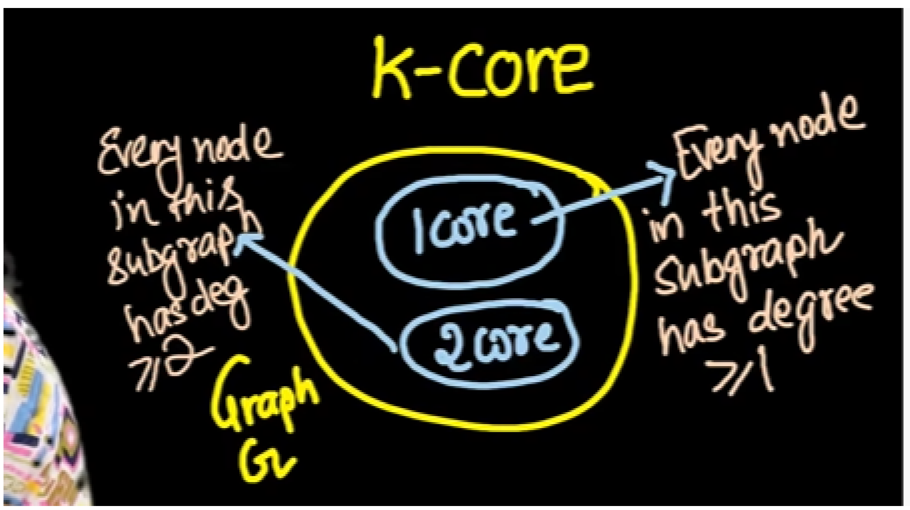


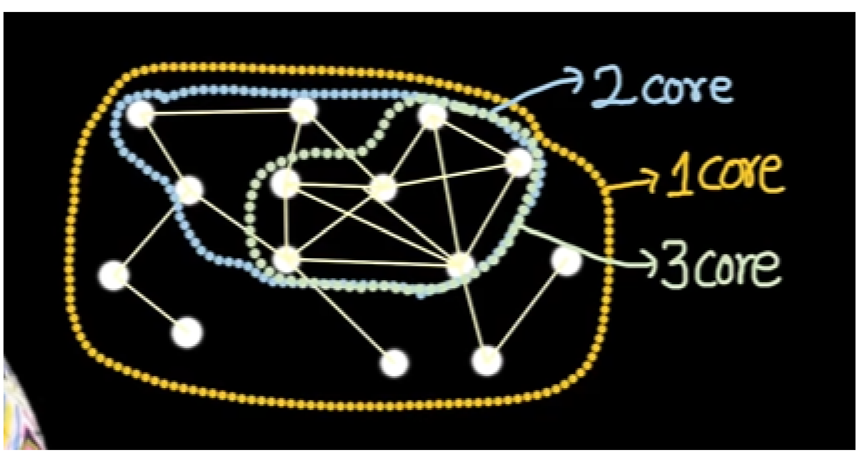
Lecture 162 : finding the right key nodes (the core)



infect the core nodes as they are densely connected and spread further

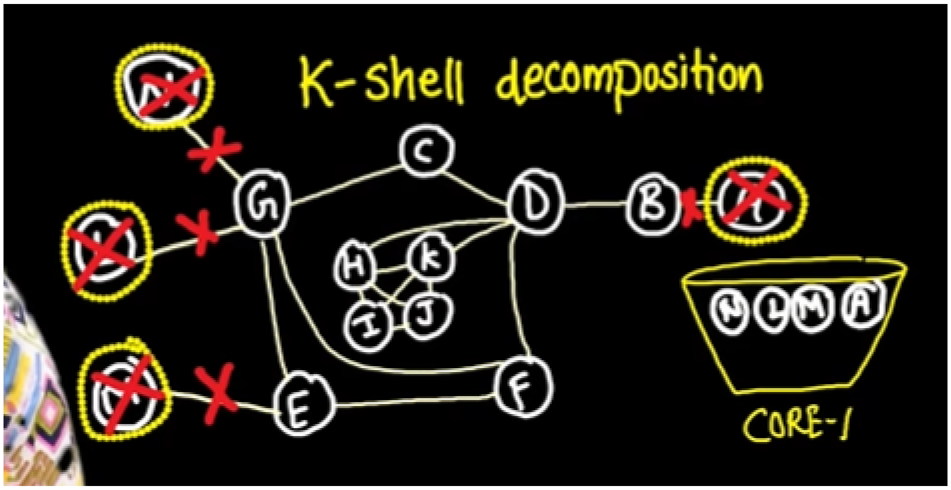
How to identify core ?



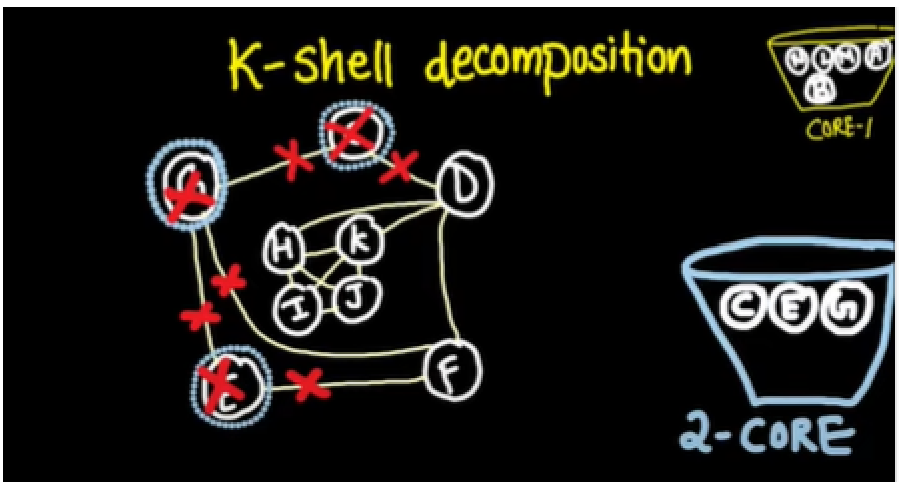


But for larger ntwrks, we use **k - shell decomposition**

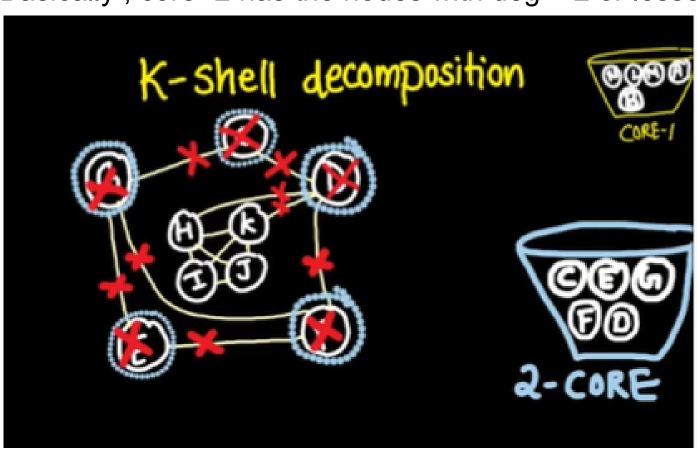
Pendant vertices - Node with degree = 1
Take the nodes with deg 1 from the ntwrk and put in core 1 bucket



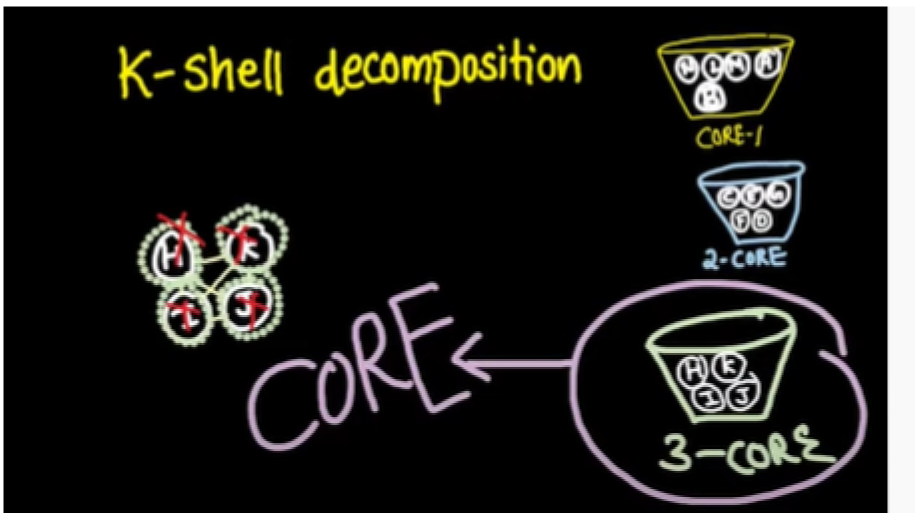
Now new nodes will emerge with deg = 1, remove it also and put in bucket.continue this till there is no deg 1 nodes
Then go for 2-core buckets

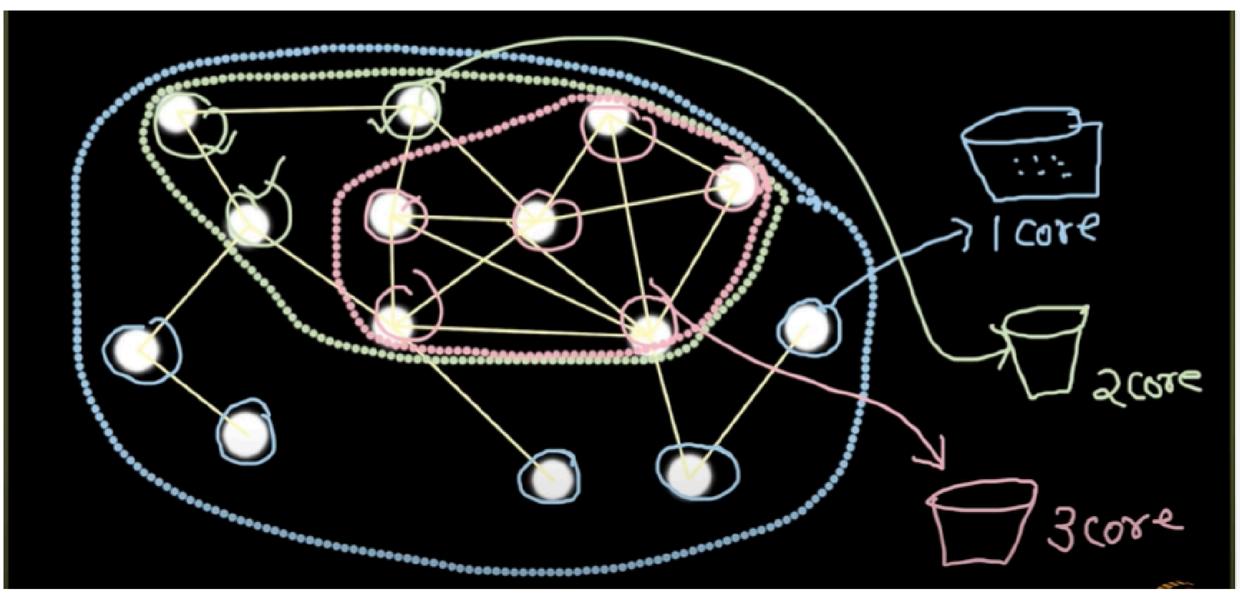


If any new node with **deg = 1 arises, remove and put it in 2-core** bucket only.
Basically , core -2 has the nodes with deg = 2 or lesser



The nodes removed at last, the last bucket = core of ntwk





The 1-core nodes of the graph is given by =

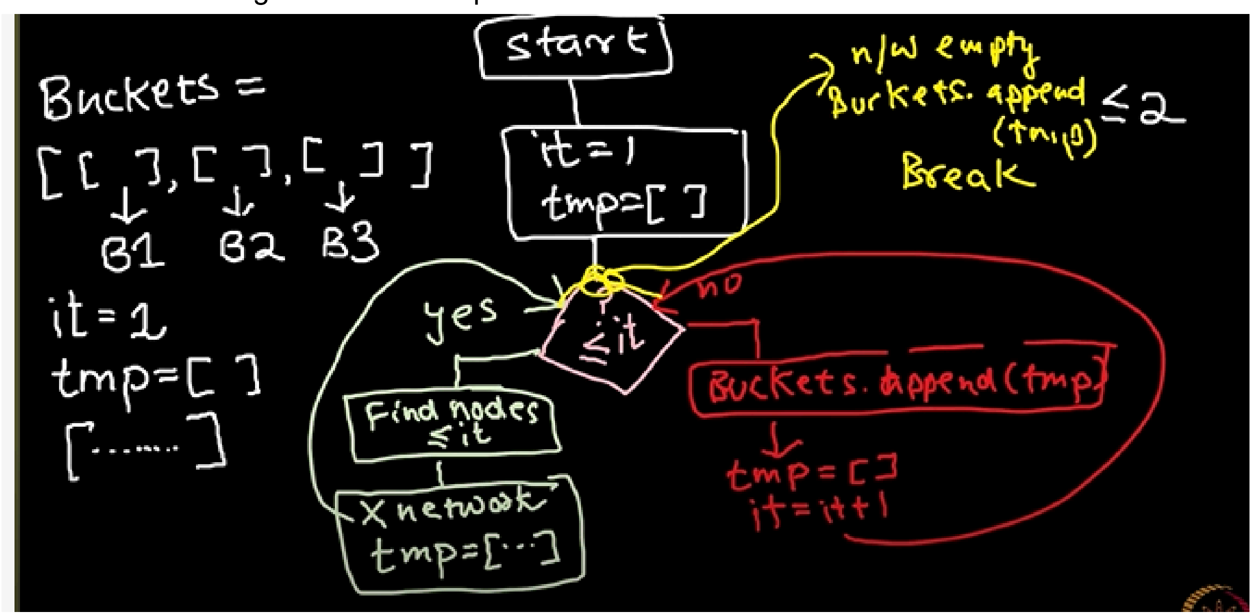
$$1\text{ core} = B1 \cup B2 \cup B3$$

$$2\text{ core} = B2 \cup B3$$

$$3\text{ core} = B3$$

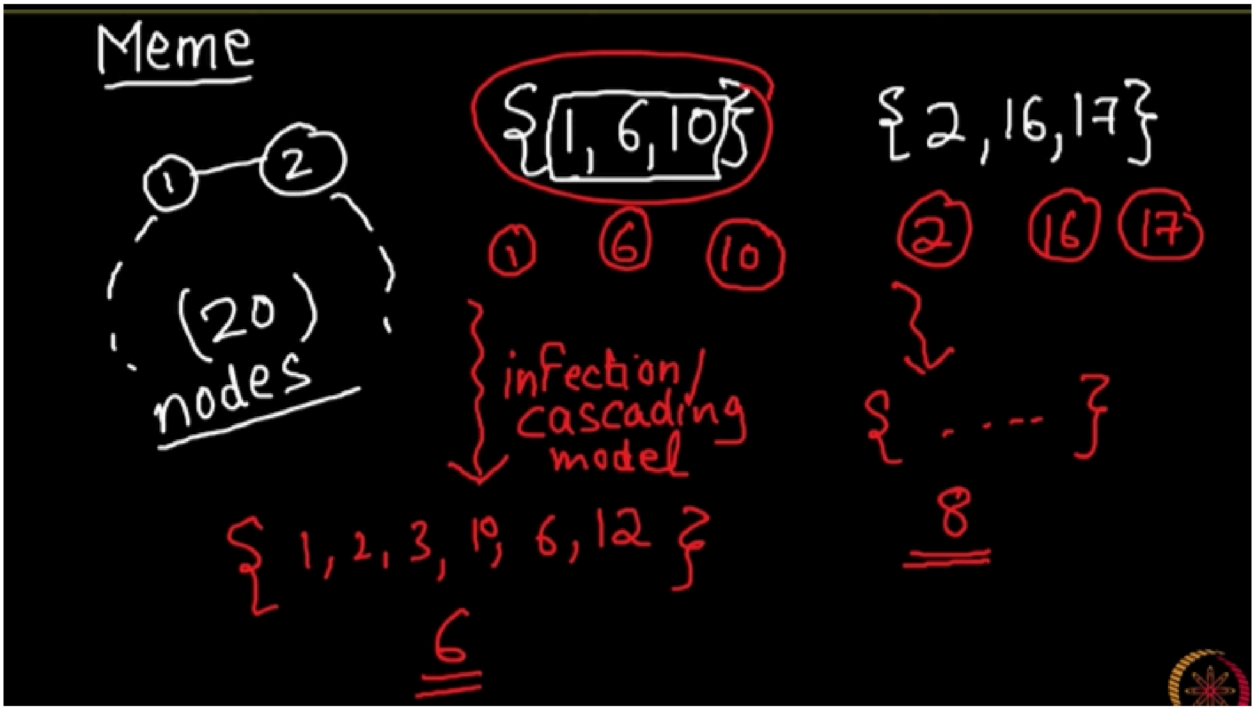
$$k\text{-core} = B(k) \cup B(k+1) \cup B(k+2) \cdots\cdots = \bigcup_{j \geq k} B(j)$$

Lecture 163 : Coding K-Shell Decomposition

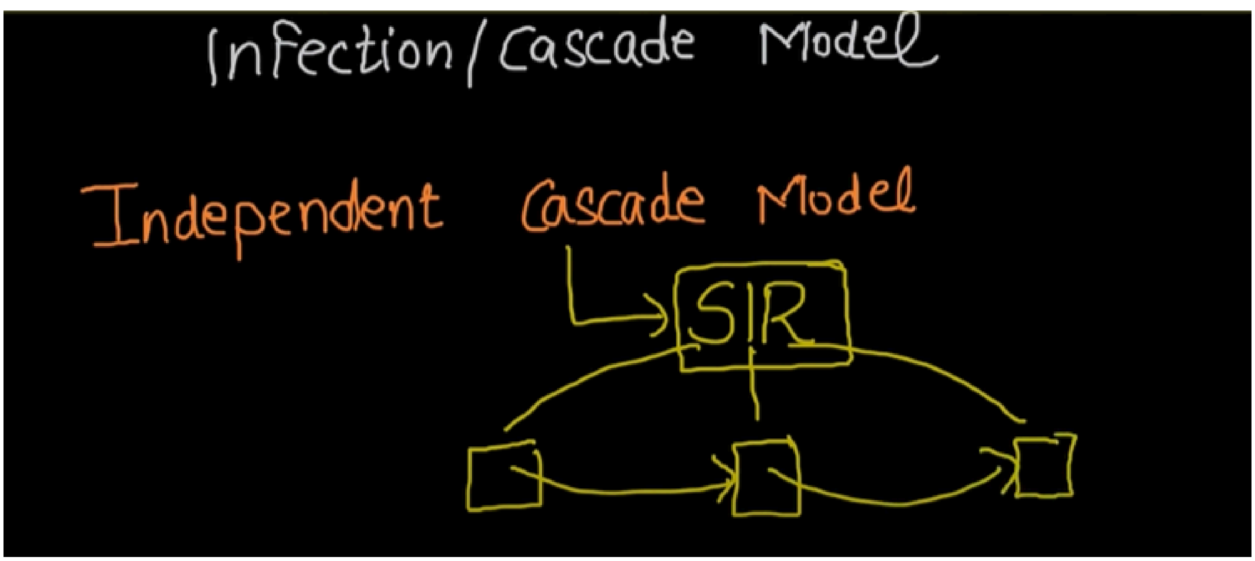


Lecture 164 : Coding cascading Model
How to know about influential nodes?

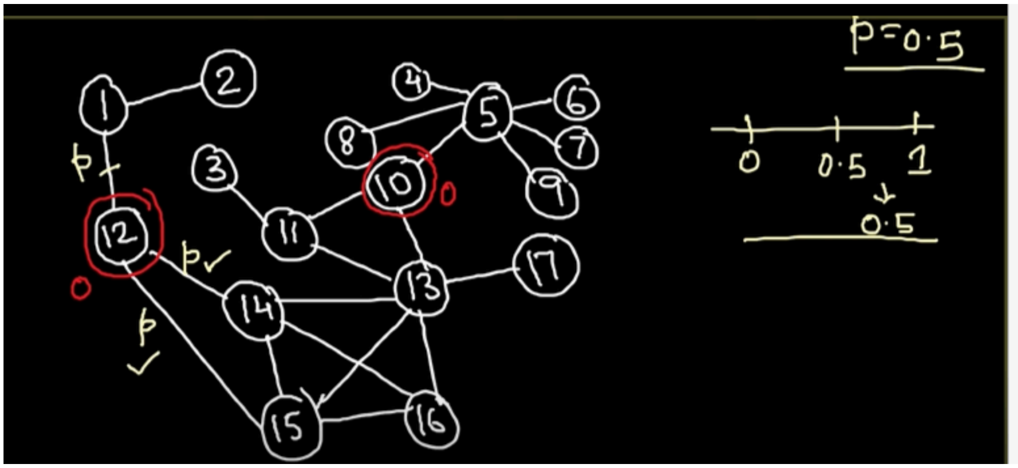
2 have 20 nodes with 2 sets of people. And u have to check which set is influential. Run a cascading model on the set and see how many total nodes get affected. This will will be **influential power**.



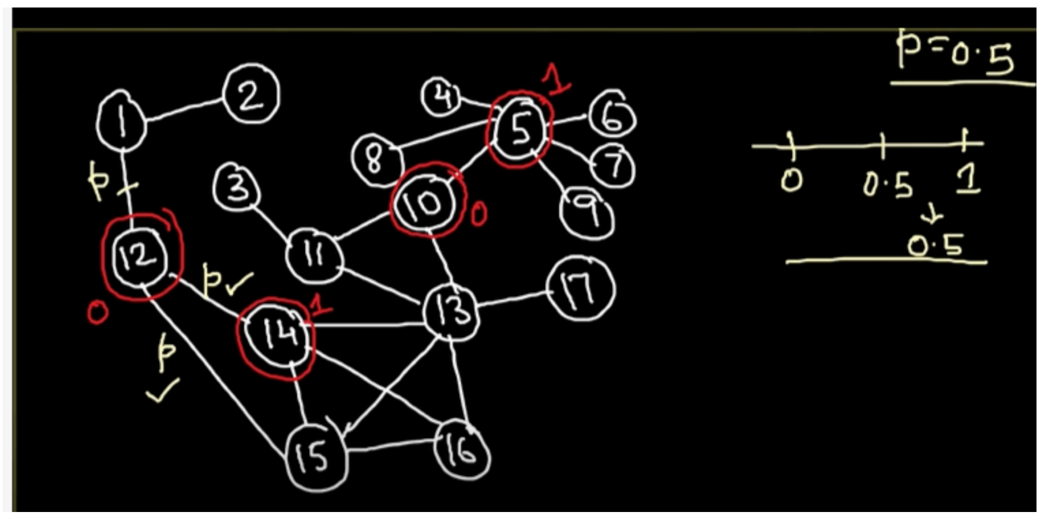
In real , we do this for all the set



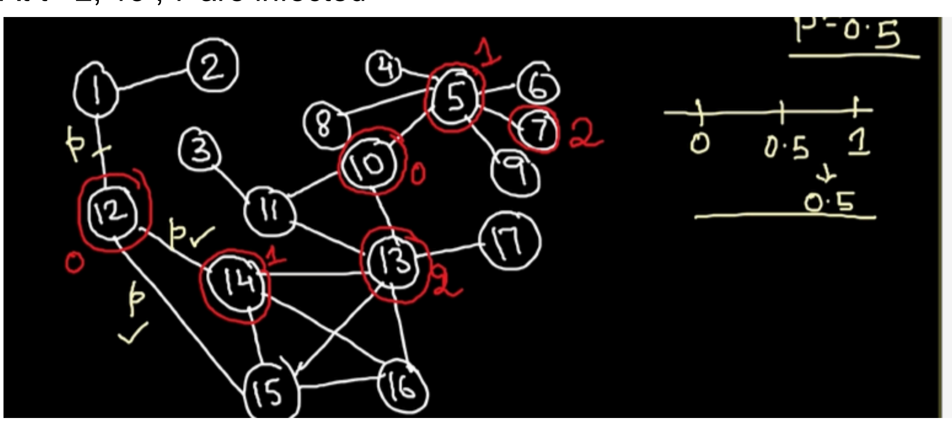
Prob for infection = 0.5



At t=0 , 12 and 10 are infected
At t =1 , 12, 10 infect others letsay, 14 and 5 got it



Now 12, 10 dont get the chance to infect again. 14, 5 can infect now
At t= 2, 13 , 7 are infected



Letsay 13 and 7 coulnt infect anyone ⇒ process stops
Process stops whn u reach an infections where noone is infected

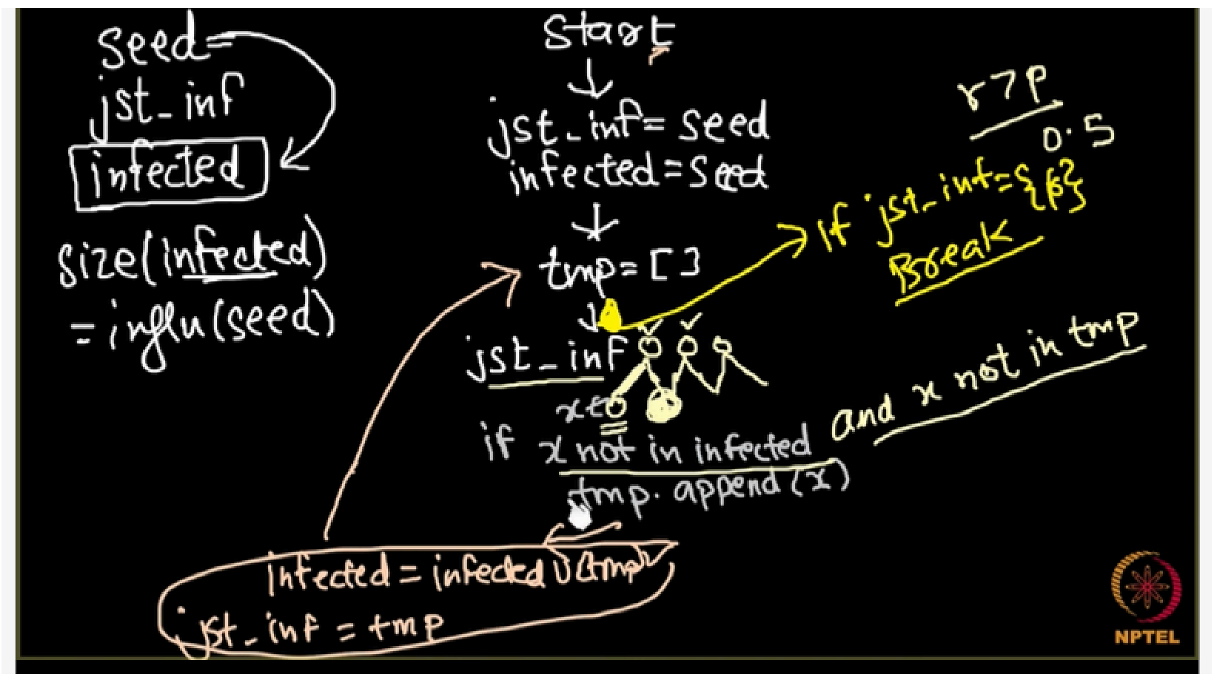
$\boxed{12,10} \longrightarrow \underline{6}$ influential power of this set = 6

And why is it called an independent cascade

model is actually very clear ah whether this
edge over here get passes the infection or node

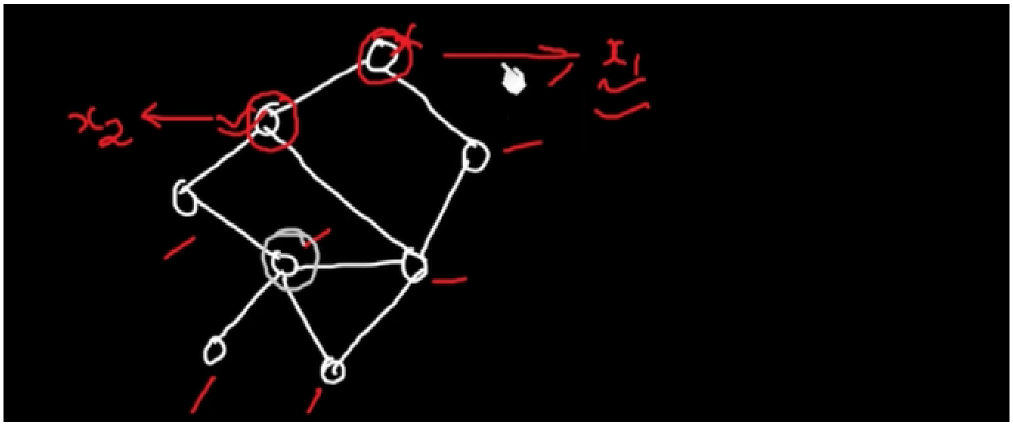
is independent of whether this says over here
passes the infection or not. So, hence it is

Coding

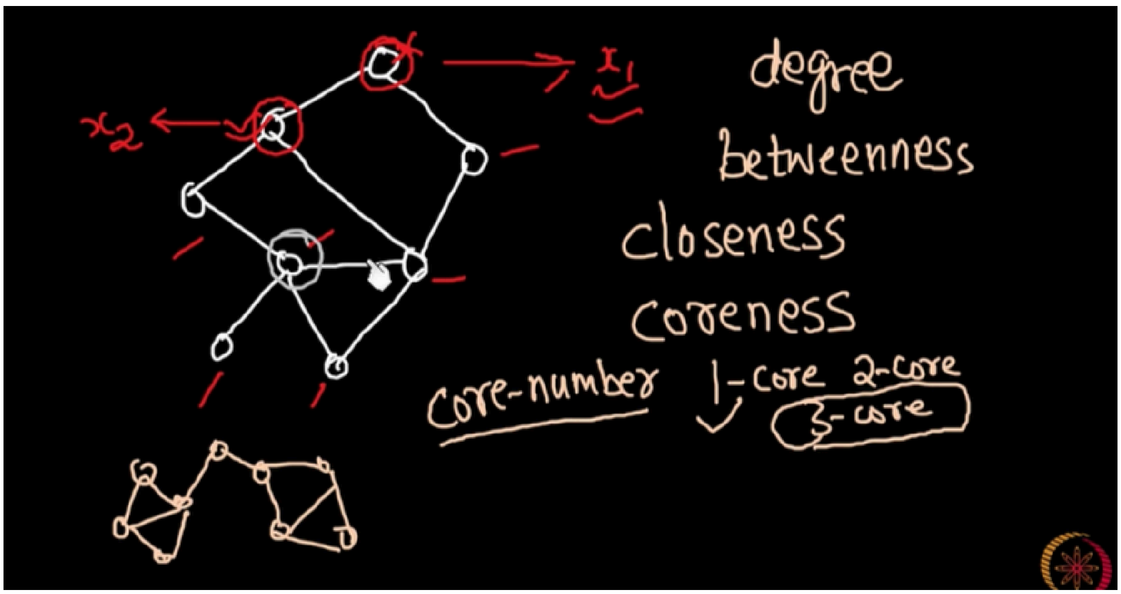


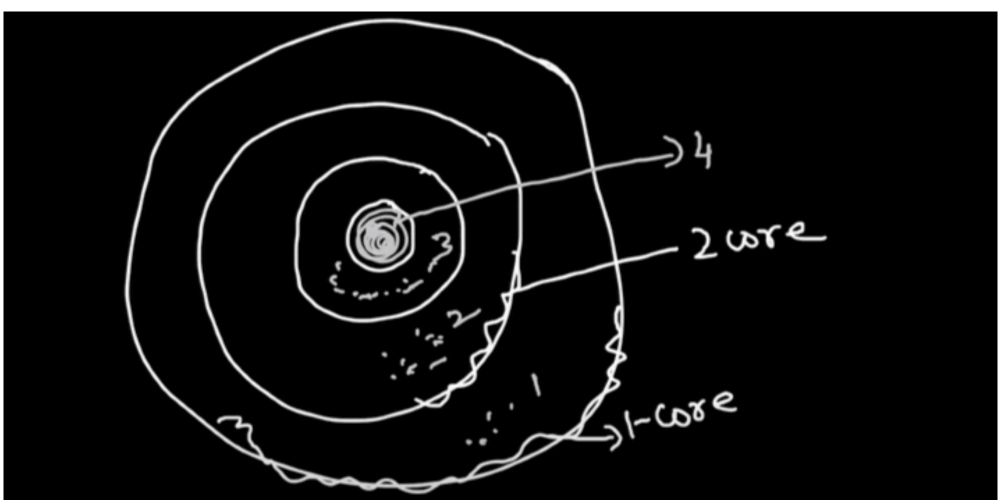

```
[3, 8]
[3, 8] [3, 8]
[5] [3, 8, 5]
[4, 7, 9] [3, 8, 5, 4, 7, 9]
[] [3, 8, 5, 4, 7, 9]
```

Lecture 165 : Coding the importance of core nodes in cascading
Calculate the influential power of all nodes taking one as seed and tell most influential - very time consuming

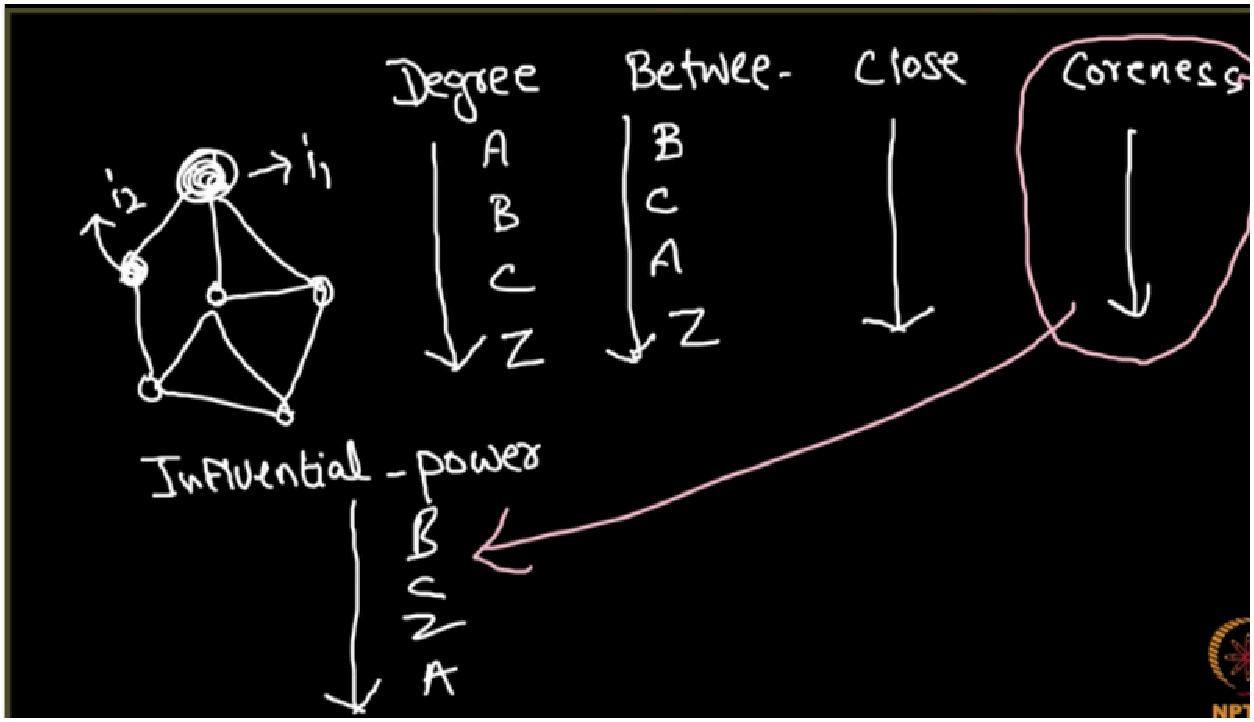


Rather u can observe and study and predict by -





Take graph, and arrange in descending order and calculate central measures and influential power. Then compare which gives the closest results



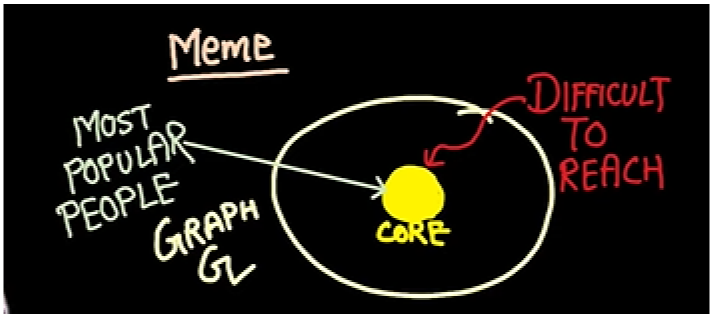
1	Degree	Closeness	Betweenness	Coreness	Influence
2	5	13	8	13	13
3	13	10	10	14	15
4	14	5	11	15	14
5	15	11	12	16	16
6	10	14	13	10	10
7	11	15	14	11	11
8	12	16	15	12	5
9	16	17	16	1	12
10	1	12	17	2	17
11	2	3	3	3	1
12	3	4	2	4	3
13	4	6	7	5	4
14	6	7	1	6	9
15	7	8	9	7	6
16	8	9	6	8	7
17	9	1	5	9	8
18	17	2	4	17	2
19					

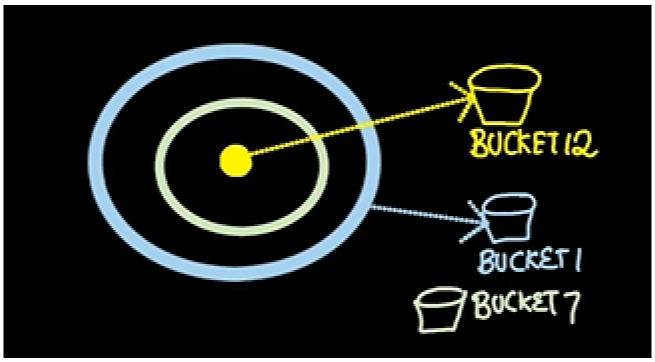
influence == cascadeness

It hass the highest similarity with **CORENESS AND DEGREE.**

AND MOSTLY , FINALLY RELATED TO CORENESS

Lecture 166 : Pseudo core

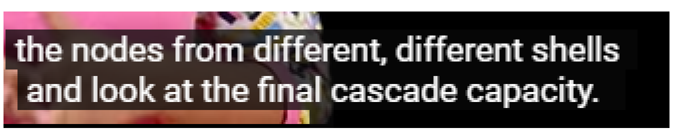


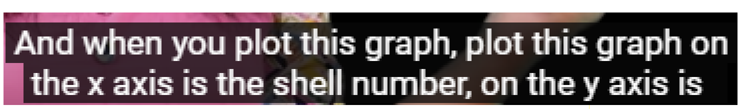


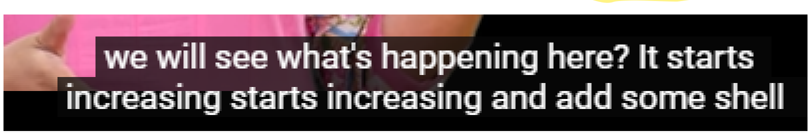
We try to know the influential power from all the shells

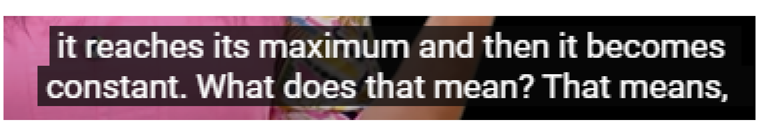
Choose innermost shell, and check how many nodes were infected. Then 11th and check for all shells
This is known as the cascade capacity of the shell

If u choose a graph, do k shell decomposition, and calculate the cascade capacity of each shell. -









hat to infect this network to make your meme go viral, it is not always necessary for you to go to the innermost core and convince these people. So, if you look at some of these shells outer from this core shell, some of these they have actually the same cascade capacity as the core nodes. So, instead of these core nodes we have found a lot more people here which are actually we can called pseudo course, we call these people as pseudo course. So, there is a big chunk of the pseudo core people, whom you can actually infect to make your meme go viral.