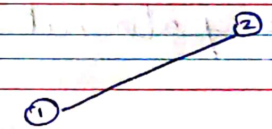
Social networks

```
import networkx

G= networkx.Graph() # undirected graph.

G.add_nodes(value) #add nodes in the graph

G.add_edge(value,value)
```

If the value given is not present in the graph, it adds the node by self.

```
G.add_node(1)
G.add_edge(1,2)
networkx.draw(G, with_labels=1)
```



G = nx.DiGraph() creates a directed graph.

```
nx.spectral_layout(G)

nx.circular_layout(G)
```

```
for u in G.nodes():
    for v in G.nodes():
        print (u,v, nx.has_path(G,u,v))
```

This prints all the nodes from one dest to another with the edge flag.

```
print (nx.dijkstra_path (G, 'node1', 'node2'))
```

This prints the shortest distance ie., dijkstra path.

```
nx.dijikstra_path_length (G, '', '')
```

If there is no path, it will return exception.

Ranking nodes in a network:

Given a bunch of webpages which is the most powerful webpage? (All webpages are interconnected)

The link with most number of inner link is the most powerful webpage. Google pagerank algorithm uses this concept.

Searching in a network:

It takes only $\log n$ steps to search in n nodes.

It takes an average of 10 steps to search in 1 BILLION nodes.

"Small world phenomena" is used to search faster.

Link prediction:

Predicts the links prediction. People you may know in facebook.

Introduction to datasets:

Zachare Karate network
34 nodes
78 edges

Connected graph - We can reach any node from a given node. This applies for all the nodes in the graph.

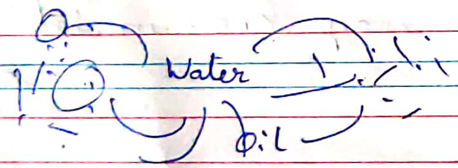
Datasets → can give answers to our random questions on the network

Example datasets:

Example 1:

Dish 1 Dish 2

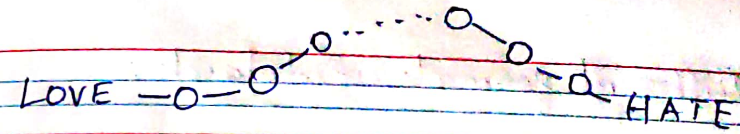
Ingredient 1 Ingredient 2



Example 2:

All words in a dictionary

This network is connected.



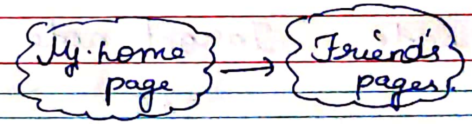
According to graph theory, theoretically there is a path that somehow connects LOVE and HATE in the word dictionary graph

LOVE is synonymous to HATE

There is degredation in synonymous when moved.

Example 3:

Web graph.



Each page is a node.

Example 4:

Facebook friends

Different formats of datasets:

CSV format
GML format
Pajek net format
GraphML format
GEXF format

CSV: Comma Seperated Values
Extensions : .txt , .csv
Edgelist format
Adjacency list format

Edgelist type

Source node	Target node	Weight
0	345	0.3
0	296	0.6
0	316	0.9
1	241	1
1	232	1.5
1	314	3.2

CSV format → string weight cannot be added.

Simplicity > Flexiblity

Adjacency list:

Source node	Adj List
1	2 3 4 5
2	3 4 5
3	2 1
4	5 1
5	3 2 4

GML: Graph Modeling Language
Flexiblity in labeling, assigning attributes and edges

```
graph
{
  node A
  {
   id A
  }
  node
  {id B
  }
}.
 edge
 {
  source B
  target A
 }
```

replace { } by [ ]

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

To assign attribute values, use graphics keyword.

Pajek net format : uses .net extensions (or) .eg

```
* Vertices no-of-vertices-in-the-network
1 "label1"
2 "label2"
3   :
4   :
:   :
:   :
n "labeln"
* arcs
source node target node
:
:
:
```

If there is no label:

```
* Vertices 10
* edges
1 2
2 4
3 2
:
```

Attributes can also be added.

* arcs

42 43 5

GraphML format:

Extension: .graphml
Uses XML tags.

```
<?xml>
<graphml>
    <graph>
    <node>
    :
    <node>
    <edge>
    <edge>
    :
    <edge>
  </graph>
</graphml>
```

```
<?xml>
<graphml>
    <graph id="G" edgedefault="undirected">
    <node id="1"/>
    <node id="2" />
    <node id="3" />
    <edge id="e1" source="1" target="2"/>
    :
    <edge id="en" source="3" target="2"/>
  </graph>
<graphml>
```

GEXF: Graph Exchange XML format

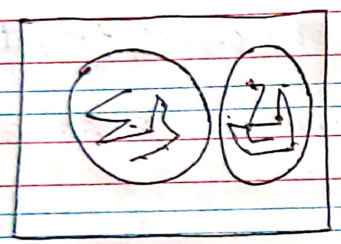
Created by gephi people.
Like photoshop for graphs

```
<geaf>
    <meta>
    </meta>
    <graph mode="static"
        defaultedgetype="directed">
      <nodes>
          <node>... </node>   and
      </nodes>                <edges>
</graph>
```

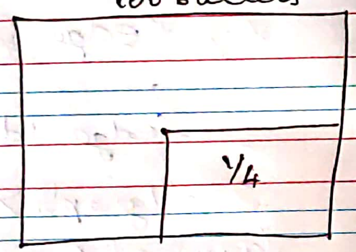
NetworkX

We can read in one format and write it in another format.

Emergence of correctedness

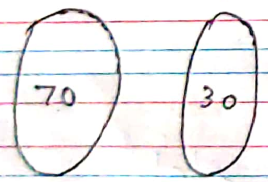


100 bullets



Its very unlikely that this place has no bullets hitting.

100 people



Total friendships for 100 people $= \binom{100}{2}$

$= \frac{100 \times 99}{2} \simeq 5000$

Total friendship across: $70 \times 30 = 2100$

Total possible friendship = 5000
Total possible across = 2100

There were 2100 possible ways in which they could have made friendship. This is not impossible but "impropable".

Assume $n=100$. Construct a graph with all possible edges. If you keep on removing an edge randomly from the graph, the graph becomes disconnected at a point of time.

Totally disconnected ——— Disconnected ——— Totally Connected

0 edges. $\frac{n(n-1)}{2}$ edges

Given n nodes as one keeps adding new edges "uniformally at random" when will the graph become connected?

Solution :

When will the graph not have an isolated vertex.

$$(1+1/n)^n = e$$

$$(1-1/n)^n = 1/e$$

Probability of V is isolated :
when an edge is included

$$P(v) = 2/n$$

$$P(v') = 1 - 2/n$$

Probability of v not isolated after k edges being added:

$$(1-2/n)^k$$

$k=1$ $(1-2/n)$

$k=2$ $(1-2/n)^2$

$\vdots$

$k=n$ $(1-2/n)^n$

$$\left.\begin{array}{l}\text{Probability of v} \\ \text{being not being} \\ \text{isolated}\end{array}\right] = (1-2/n)^n$$

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

$$= (1/e)^2$$

$$= 1/e^2$$

P(v not isolated) = ? when $k = n \log n$

$$= \left[\left[1-\frac{1}{n/2}\right]^{n/2}\right]^{2\log n}$$

$$= \left(1/e^2\right)^{\log n} = \left(1/e^{\log n}\right)^2$$

$$= \left(1/n\right)^2 = 1/n^2$$

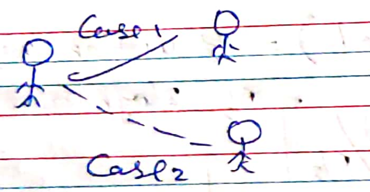
Probability of vertex V becoming isolated after including $n \log n$ edges is $(1/n^2)$

Like if $n=100$, $n \log n$ like 120

$\boxed{1/100^2}$ is this P(v not iso)

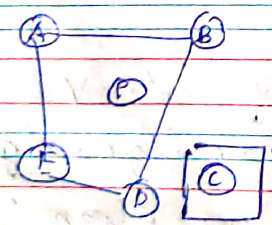
Conclusion:

If we a take a graph G with n vertices and if $n \log n$ edges are added uniformly at random there is a very low probability that there will be an isolated vertex.



Case 1: More familiar
Case 2: Not familiar

Strength of Week Ties:



C doesn't know A, B, E, D.

C is different from new world. So he may help F by giving the new information.

Week ties are actually strong. Strong ties are actually weak.

Acquaintances are very important.

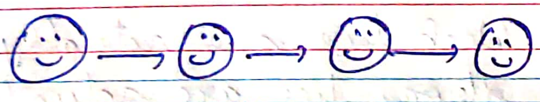
New information always comes from a different network.

21.

Week 4.

Video 1:

Happiness are contagious upto 3 levels.



If you are happy, your friends are happy → their friends are happy → their friends are also happy.

Selection	Social influence
I speak hindi. I meet a girl who also speaks hindi. This is called selection	Smoking. We influence people to drink or smoke. Generally smoking & drinking are influenced by people.
Here, the character is already the same.	Here we become the character after the influence.

Social status, Class toppers. Eating junk food, Partying

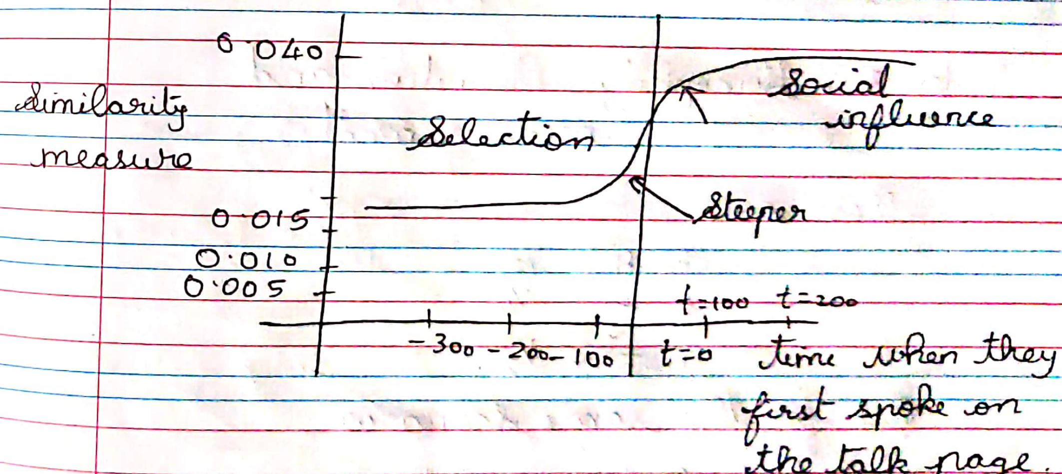
Viedo 2:

How wikipedia works?

Wikipedia is trustworthy.

$$\text{Similarity measure: } \frac{\text{Common entry (Common)}}{\text{Total entry (unique)}}$$

Rama Prem



Two people spoke because of their common interest. Two people become friends if they have a lot of common interests.

Average taken over all possible pairs of wiki editors who spoke to each other.

Inference: You select when they are similar. This is selection.

After selection, they may be hanging out together which is a social influence.

Toss twice:

Video 3: Homophily: Definition and measurement

Toss coin:

H H } 1/4
T T } 1/4
H T } 1/2.

But for the same scenario,

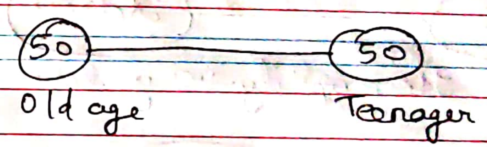
Teenager Old age
T O

T T 1/4
O O 1/4
T O }
O T } 1/2

But teenagers will like to talk only with teenagers.

But a group of men & women will be differing.

Does the network exhibit homophily? To what extent does it exhibit.



But at random if you chose,

T — T friendship has the more probability.

Here H-T, T-H probability should be 1/2 but it is very less.

Bollywood actors $\leftrightarrow$ Bollywood actress

Very less within actors — Very less within actress.

More friendship between Actors & actress

It is an opposite of the above scenario.

Old age ——— Teenage

Homophily more

Actors ——— Actress

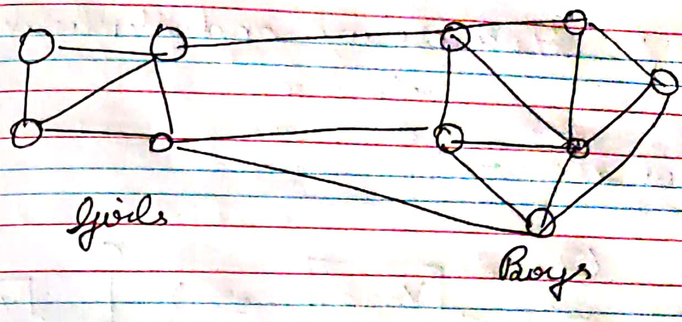
Homophily very less

Girls ——— Boys

$$\text{Homophily} = \frac{3}{9} = \frac{1}{3}$$

$$\frac{\text{Actual friendship between Boys \& girls}}{\text{Expected friendship between boys \& girls.}}$$

Expected is 9 because there are totally 18 edges (friendships) in which 1/2 should be cross gender. Hence $18/2 = 9$.



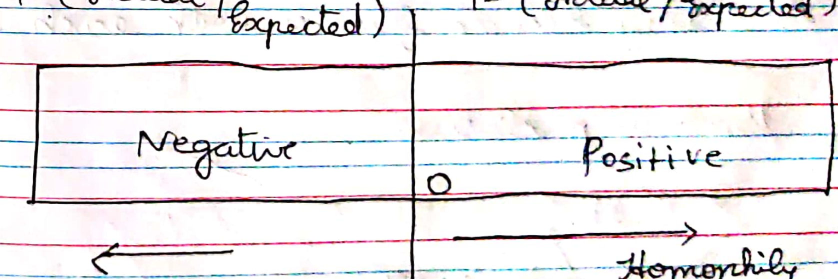
Lesser the fraction more the homophily.

$$\text{Fraction value} \propto \frac{1}{\text{homophily}}$$

$1 - \left.\text{Fraction value}\right\} \Rightarrow 0 \Rightarrow$ no homophily

$1 - \left.\text{Fraction value}\right\} \Rightarrow$ negative $\Rightarrow$ heterogeneity

1 - (Actual / Expected) | 1 - (Actual / Expected)



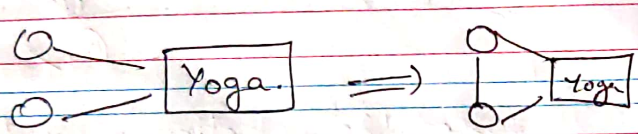
Heterophily (Different people) (Actress, actors)

Half the times diff

Homophily (Similar people) (Teen, old) ~~Actress~~

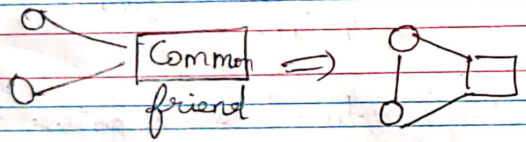
Video 4: Foci closure and Membership closure

Focal closure:



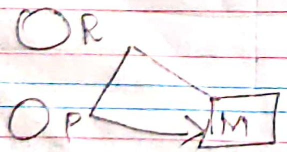
Two people have a common place where they meet and talk with each other and make friends.

Triadic closure:



Two people have a common friend and they become friends with each other.

Membership closure:



Since I am a member of a club, I will take him to the membership since we are already friends.

Video 5: Introduction to Fatman evolutionary model

Programming model for people based on weight.

Video 6:

Fatman evolutionary model
Adding people.

Step 1: Generate 100 random people.

Step 2: Generate random ~~so~~ BMI for each person.

Step 3: Display with labels and the size of the person.

Video 7:

Fatman evolutionary model
Adding social Foci

Step 4: Add the foci nodes in the graph.
(eatout, gym, ...);

Step 5: Color the graph persons & foci separately.

Step 6: Add edges to the foci nodes randomly.

Step 7: Display the graph.

Video 8: Fatman evolutionary model
Implementing Homophily

$$\text{Probability of Friendship between X and Y} \propto \frac{1}{\text{Difference in BMI} + \underline{\underline{1000}}\ (\text{nonce})}$$

reduces probability

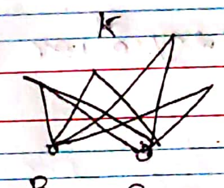
For adding edges, find the probability of the friendship with the given formula.

$$P(X \text{ and } Y) = \frac{1}{|BMI_1 - BMI_2| + 1000}$$

If the probability is less than 0.4, make an edge.

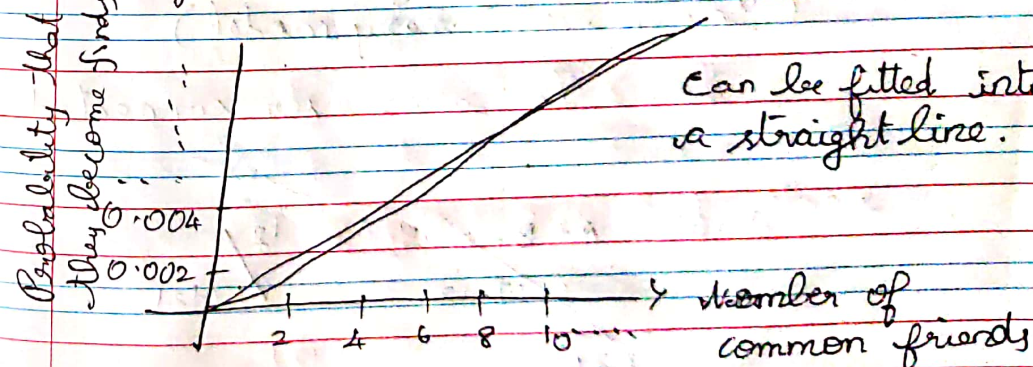
For that a random edge is generated between two nodes.

Video 9: Quantifying the effect of
Triadic closure



If B and C have k no. of common friends, there is a chance of (more chance) of B & C becoming friends.

The graph is almost linear.



Can be fitted into a straight line.

$(1-p)^k$ is the probability that B and C are not becoming friends eventhough they have k common friends.

$T(k) = 1-(1-p)^k$ is the probability that B&C tend to become friends which is the equation for a straight line.

p is very small.

$\boxed{p \propto k}$

Video 10: Implementing Closures
(Fatman evolutionary model)

Foci & Foci are never connected.

$A \cap B = A + B - A \cup B$

$21 + 17 - 12$

26

$12 / 17 + 21$

$21 + 17 = 38$

$12 / 38 - 12$

$6/19$ 26

$13/26$ $1/13$

$\frac{A \cup B}{A}$ $\boxed{\frac{A \cap B}{A \cup B}}$

Week 5 Video 1

Spatial segregation

We tend to be surrounded by people who are likely you.

R	G	G
R	G	R
R	G	R

$t = 3$

When the threshold is less than 3, we will stay. If it goes ~~beyond~~ below 3, then we migrate.

Unhappier people change their location to happier place.

Schelling model ⇒ simulates this scenario.

In a grid you are always surrounded by 8 people.

Video 2

Simulation of schelling model

25% similar $\Rightarrow$ $8 \times \frac{25}{100} = 2$

$8 \times \frac{25}{100} = 2$

50% similar $\Rightarrow$ $8 \times \frac{50}{100} = 4$

75% similar $\Rightarrow$ $8 \times \frac{75}{100} = 6$.

You should be surrounded by atleast 6 people.

Segregation takes place gradually.

$\boxed{\text{Similarity} \propto \text{Segregation}}$

Video 3: Conclusion

Emergence of homophily:

Segregation is not really needed sometimes. Eventhough the people are scattered, they are happy.

Video 4: Schelling model implementation

Satisfied node - neighbors are more than t

Unsatisfied node - neighbors less than t

Video 5: Schelling model

Base code) + Video 6

```
G = nx.grid_2D_graph(N,N)

pos = dict((n,n) for N in G.nodes())

labels = dict(((i,j), (i*10+j)) for i,j in
                G.nodes())

nx.draw_networkx_labels(G, pos, labels)
```

Add diagonal edges,

```
for (u,v) in G.nodes():
    if ((u+1<=N-1) and (v+1<=N-1))
        G.add_edge((u,v), (u+1, v+1))

for (u,v) in G.nodes():
    if (u+1<=N-1 and v-1>=0)
        G.add_edge((u,v), (u+1), v-1))
```

Assign random type to the graph:

```
for n in G.nodes():
    G.node[n]['type'] = random.
                         randint(0,2);
```

Maintain all types of people in a list:

```
type1_list = [n for (n,d) in G.nodes(data=True)
                if d['type'] == 1)
```

Similarly for type2, type0.

To color the graph:

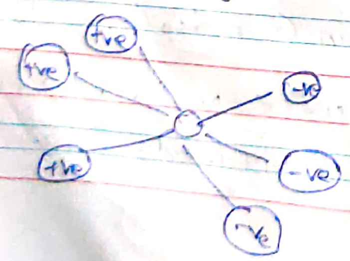
```
nodes_g = nx.draw_networkx_nodes(
    G, pos, node_color='green', nodelist=
            type1_list)
```

Video7:

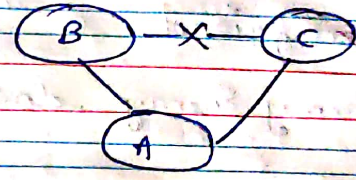
Getting a list of unsatisfied nodes

Video: 8

Positive and Negative relationships

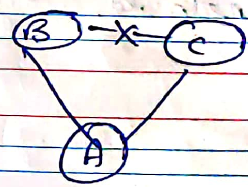


Video: 9 Structural Balance

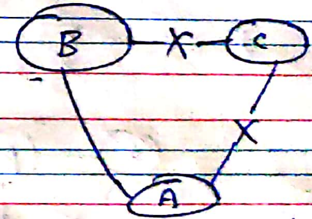


A and B and A and C are thick friends.
But B & C do not talk to each other.

Hence, there is a problem when A, B, C unite together.



Unstable situation



Stable situation

For three friends:

A – B – C – A

A – X – B – C – A

A – X – B – X – C – A

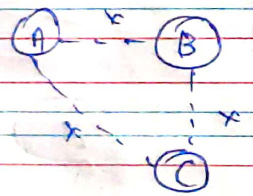
A – X – B – X – C – X – A

(Enemy's enemy is an enemy.) ⇒ Wrong.

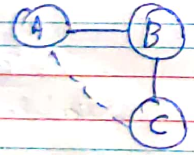
Video : 9. Enemy's enemy is a friend

Enemy Enemy Enemy does not exist.

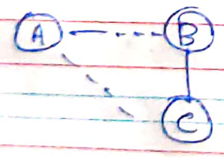
A and B will be nice friends.



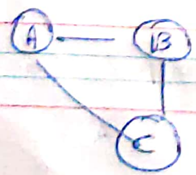
Unstable.



Unstable



Stable



Stable

Video 10 : Structure of balanced networks

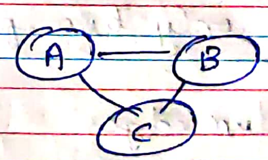
30 people : $^{30}C_2 \simeq 30^2/2 = 450$

If the given graph is unstable, the graph moves towards stable state.

What will be the result of the stable state ?

In a group of positive stable relationship, when we introduce one negative state, it spreads along the whole graph.

Video 11 : Balance theorem



→ Chaos when all three fights each other.

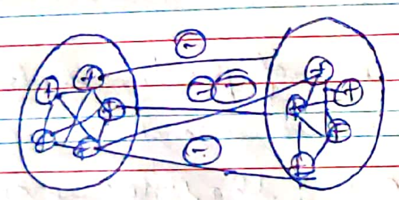
No matter how much we put for a fight, it is always two people who fight.

Base : Enemy's enemy is a friend.

Example : World War I, II

Video :12 Proof of Balance theorem

Eventhough you try to create a graph with stable relationships you end up forming clusters where the +ve are inside the cluster and the -ve are outside the cluster.



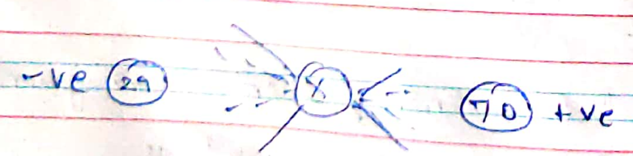
This means there is always two sets who fights.

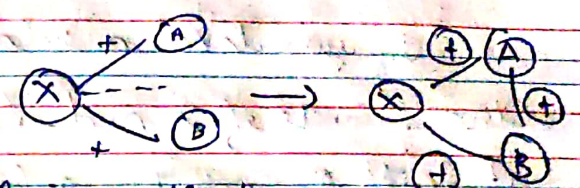
Here there are only two possible structures:

① All positive edges

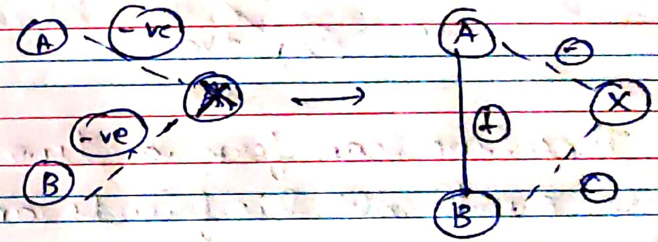
② Clusters with negative edges only across other clusters.

Proof:





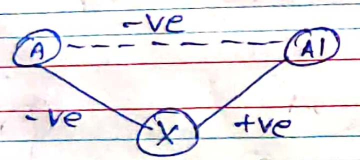
It is obvious that in order the graph should be stable, A and B will have only ⊕ relationship



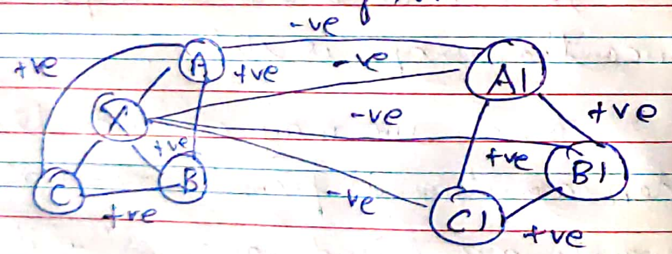
Here, A and B should be positive, since the graph is stable.

Hence, we may conclude that when an item is picked where all of its elements are friends and all of its elements are negative, you will observe amongst the positive friends, there will be positive relationship and among the negative relationship, there will be ~~negative~~ positive relationship only.

If you take a relationship between the friends of X and enemies of X, it will be negative relationship in order the graph must be stable.



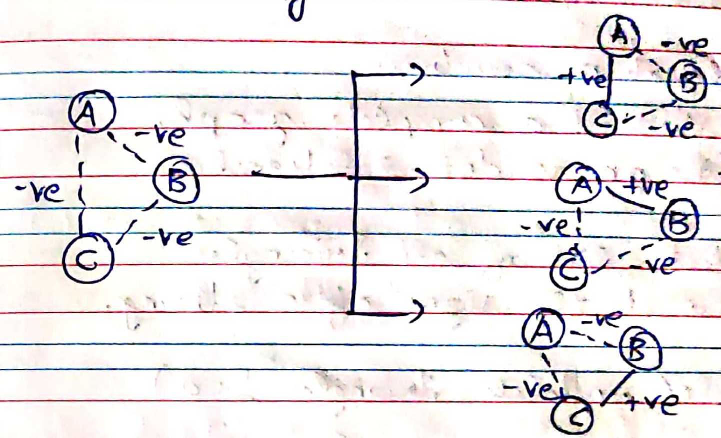
Hence, we can form a two clusters here. X with all his friends and a cluster with enemies of X.



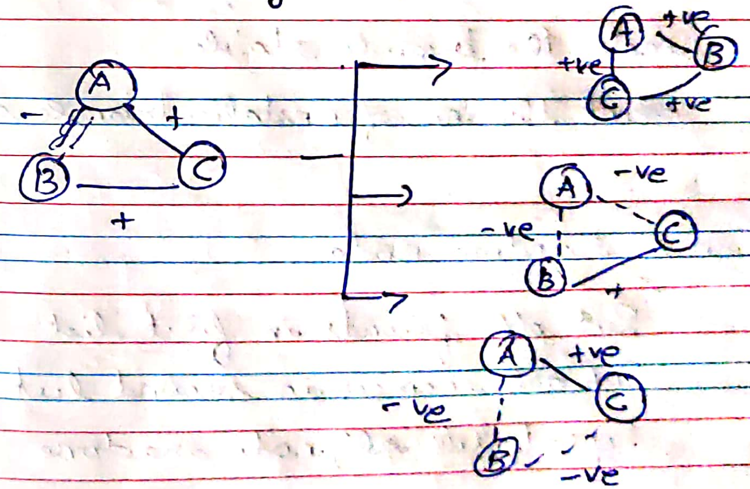
Hence, this proves that there are only two possibilities.

Video13: Signed networks (code)

Unstable triangle 1.



Unstable triangle 2



Video 14: Outline of implementation (Code)

Countries relationship.

Create n countries.
Make it a complete graph.
Assign random relationships

Get list of all triangles
Count the signs after storing.

```
while( unstable_triangles != 0 )
{
    chose a triangle which is unstable
    make the triangle stable
    count the no. of unstable triangle again
}
```

```
Chose a random node
{
    Put all friends in first list
    Put all enemies in second list
    Repeat till all nodes are done
}
```

Video 15: Creating graph, counting unstable triangles

Get all list of triangles in a graph:

```
trust_list = [list(x) for x in
             itertools.combinations(nodes,3)]
```

$\boxed{\text{nodes = G.nodes()}}$

Count the unstables

Get signs of tris

```
↳ for i in range (len(tris_list)):
      temp.append (G[tris[i][0]][tris[i][1]]
                   ['sign'])

      temp.append ( 1 , 2 )
                  ( 2 , 0 )
```

Count the unstable:

```
for in range (len(all_signs)):
    if( all_signs [i]. count('+')==3 )
              or 0
        stable +=1
    else
        unstable +=1
```

Video 16 : ~~Counting~~ Moving unstable to stable state

```
while (unstable != 0)
{
    G = move_a_triangle_to_stable()
    all signs = get_signs_of_tris ()
    unstable = count_unstable ()
}

move_a_triangle_to_stable ():
    found_stable = False
    while (found_stable == False) :
        index = random.randint ( 0, len(tris)-1 )
        if (all_signs[ index ].count ('+') == 2
                or 0)
            found_stable = True
    else
        continue

r = random.randint (1, 3)

if( all_signs [index].count ('+') == 2):
    if (r == 1):
        if (G [tris [index] [1]]
```

Change the signs in the possible way that was shown before.

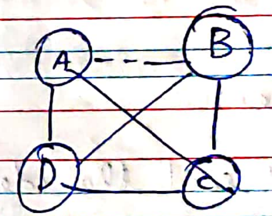
Video 17 & 18 Forming two coalition

Breadth first search is done to be added to the friends list.

All the enemies are added to another list

Video : 19 : Visualizing :

Graphs are coloured according to the friends and enemies of the random node chosen.



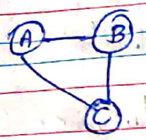
ABC

ABD

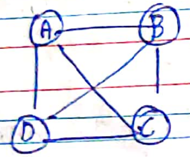
ACD

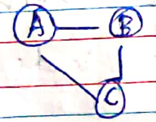
One unstable triangle → even positive edges. Hence the graph will change.

$$\frac{n(n-1)(n-2)}{6}$$



④ $\frac{3(3-1)(3-2)}{x\,6} = 1$





ABC ACD

ABD ~~BA~~ BCD

C

$\frac{4(3-1)(3-2)}{6} = 4$

Week 6 : The web graph

Hyper link used to connect and take you to other link.

To find out best pages:

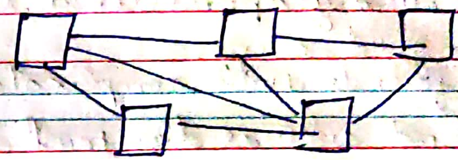
Hire 1000 people and sort the idea with the help of keywords and rate it.

Now display the web page which is good and in top of the table.

What google days does:

This is more more better way.

A web graph is a directed graph which connects all the web pages in the graph with the help of hyperlinks.



Video 2: Collecting the web graph.

1) How to collect web graph
2) How the web graph will solve our problem

$\boxed{nTh}$ A city and a given car, you would have explored almost all the paths in the graph within a month.

Generate a random graph
Pick a vertex randomly
Start walking on the graph randomly
When will one visit all the vertices?
Will all vertices be covered ?

<-- Python code explanation -->

Video 3: Equal coin distribution

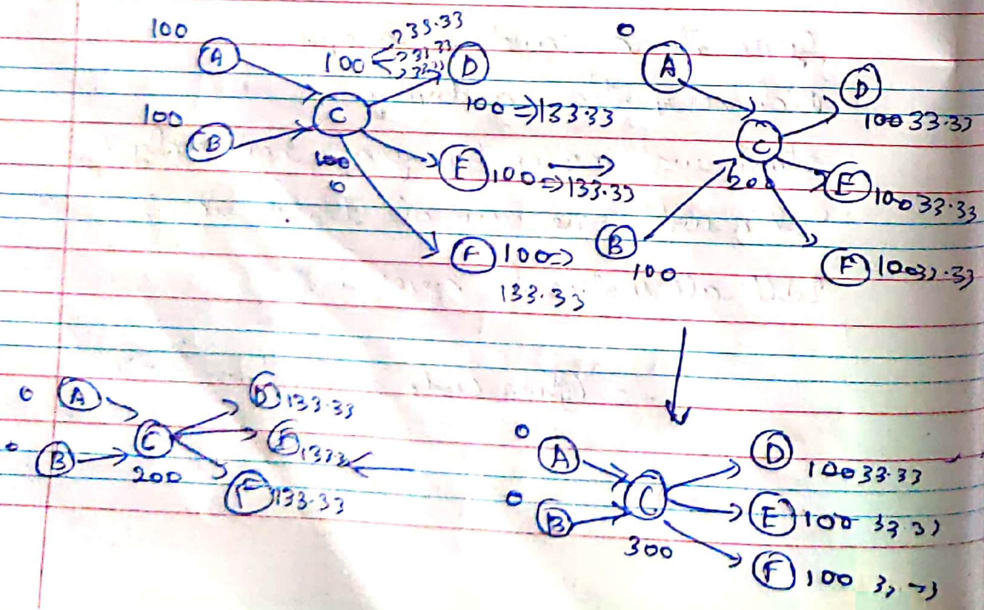
How the web graph is used to solve?

given 30 friends with complicated edges?
How to find who is the most important?

The strategy used to find the most important person here is:

Allocate some gold coins to each vertex.

Distribute them equally to the outgoing edges and the vertex will get coins from web pages which are pointing to it.
(friends)



Now C, is the most important node in Graph.

Video 4: Random coin Distribution

Strategy used:

Start with a node.

Randomly pick a neighbor and give him a gold coin.

From that node randomly pick another neighbor and give him a gold.

Repeat this process a million times and check the node which has the maximum gold coin.

It will be same as that of the previous experiment.

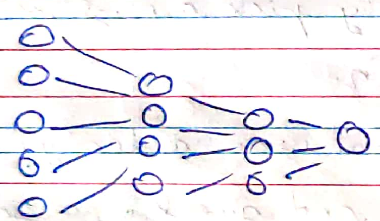
Every second, third .... but also matches.

Equal sharing = Random dropping

A node will get higher number of coins only if they have many edges point to it.

For me getting a lot of gold coins, people that are friends to me should have a lot of gold coins.

This is a recursive procedure and it goes on.



Hence, a rich node can be defined as → many nodes point to the node and there are only few nodes outgoing.

A node will be highly ranked, if it is pointed by many highly ranked nodes.

Video 5: Google page ranking

When I search for a keyword, it looks for the keyword in all the webpages that are available and allocate gold coins to each of those websites.

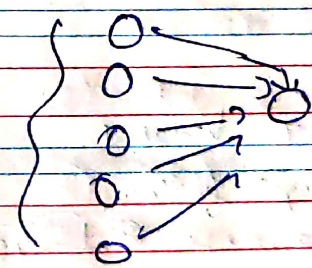
And it then displays the results in the descending order to display the page which has the maximum gold coin.

Video:
PageRank vs DegreeRank

High in degree but their page rank is low. But why?

Because, the in degree vertices are not so important.

more non imp pages



→ makes the
make non important

Low in degree, but highly ranked? How?

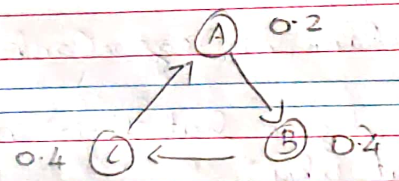
Because ~~those~~ that pages ~~are~~ is ~~linked to~~ referenced by most important pages.

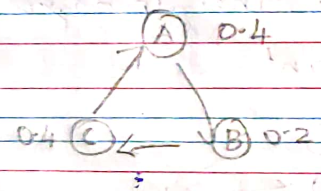
less imp pages

→ makes important

Hence, there is no relationship between the degree and the rank of a web page.

$\boxed{\text{RANDOM WALK ALGORITHM}}$





Week 7 Video 1

People follow

You like pizza more. But everyone is eating burger. So, you will also do the same. You will also be forced to eat burger.

We do not take a decision all by ourself. We get feedback from various people.

Video 2: Why do we follow?

When everyone wrote a wrong statement, we also forced to do the same which is not appreciable.

Hence following is not always good.

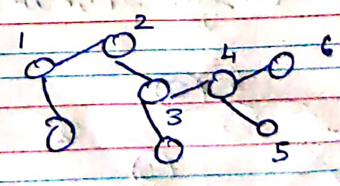
Explicit benefit : We get some benefits when we force others to follow.

Informational benefit : Information gain when we follow others.

Video 3: Diffusion network

If entire people are considered as nodes and there is an edge if they talk each other.

If a product is made to sell in this network, it kind of diffuse through the network



It depends on the product.

Video 4: Modeling diffusion.

We have two choices. Go to library / Enjoy the day in shopping.

Now, we have 20 friends out of which 18 friends tend to go to library and they invite you.

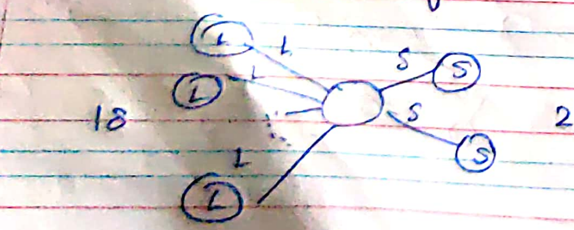
Now, we might think to go to library.

Payoff

Library < Shopping

A	L	S	L	S
B	S	L	L	S
	X	X	L	S

Obviously the payoff for S is higher since there is much fun involved.



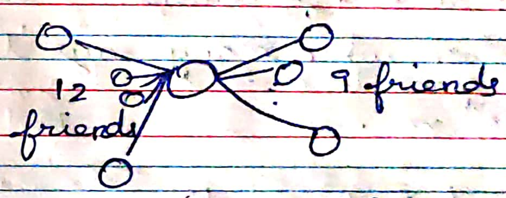
The payoff you get in library is $18 \times L$ and the payoff you get in Shopping.

Suppose $L = 2$ & $S = 3$.

$18 \times 2 = 36 \quad 2 \times 3 = 6$

$L > S$.

Hence, you will go to library.



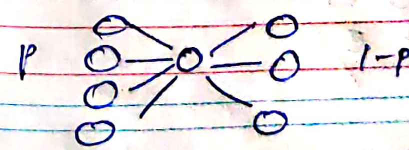
$L = 2 \quad S = 3$

$12 \times 2 = 24 \quad 9 \times 3 = 27$

You will go to shopping.

Video 5: Modeling diffusion (Continued)

A (a) B (b)



p fraction of friends.

$A \Rightarrow p * a$

$B \Rightarrow (1-p) * b$

If A has to be adopted,
$pa \geq (1-p)b$.

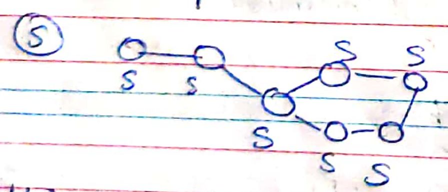
If
$$p \geq \frac{b}{b+a}$$

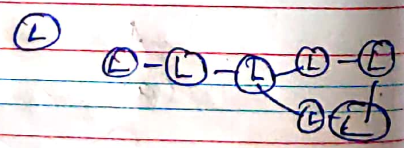
We have a threshold. 40% => I will take side A.

Social Reinforcement => We belive the information, when many says the same information.

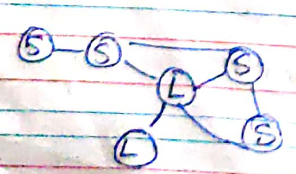
40 friends are there. 2 nodes change their mind. Some see these 2 nodes and change their mind. This spreads along the network.

Three possiblities





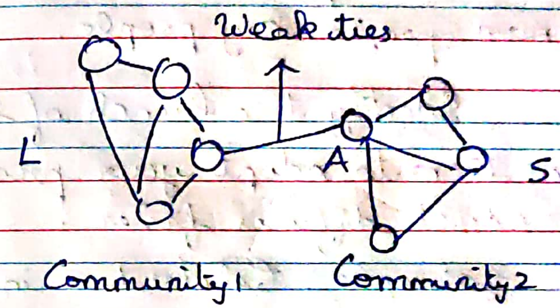
Mixed:



Video 6: Impact of communities on diffusion:

Scenario 3:

One part ~~is~~ fully go to library, one part ~~is~~ fully go to shopping



Here convincing the node A to go to library is difficult because most of his friends are willing to go for shopping.

Hence each community has a trapped decision.

We need only few people to adopt it, and then the product diffuses easily into the network.

How do we solve this starting trouble People are not willing to change the existing product.

Example : Windows and a new OS

Scientist with a time machine calls a volunteer.

Solution : Increase the payoff - ①

New product is influenced by the ~~common people~~ key people.

Identify the bunch of key people. Force them to adopt your product. Now, the product diffuses faster.

Example : Celebrities in AD.

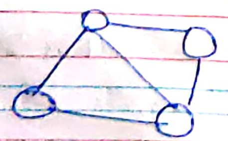
Density of a community tells you how well the ~~of~~ community is connected.

So the density affects the diffusion.

$$\text{Density} = \frac{\text{No. of actual edges}}{\text{Total no. of edges possible}}$$

$$= \frac{5}{{}^5C_2}$$ 5 - number of nodes.

$$\frac{n(n-1)}{2}$$



Farmer - 3 sons - Wood sticks story.

If the community is strong, you cannot convince them to change their idea.

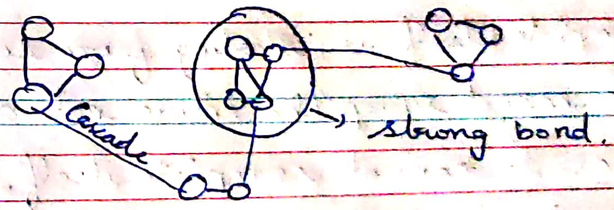
Asking Vimal to come out from the girls community is almost impossible. Hence,

$$\text{Density} \propto \frac{1}{\text{Diffusion}}$$

Video 7 : Cascade & Clusters

Lets split the entire class into small small clusters.

The trajectory of diffusion is called a cascade.



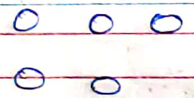
Let we denote the threshold by q.

$$p \geq \frac{b}{b+a}$$

ⓠ → Threshold

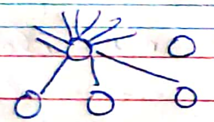
q is associated with each node.
We assume same threshold value.

Cluster density: If you look at every node in the cluster, atleast D fraction of these nodes friends is in the cluster itself.



5 people in the cluster.

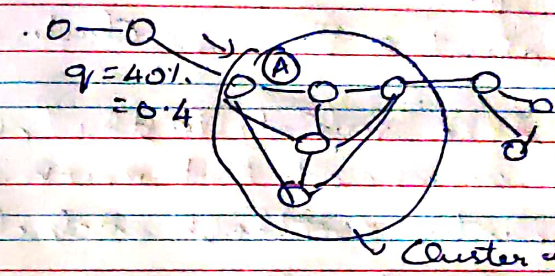
30% of the friends are in the cluster for the given node. $D = 0.3$



Theorem: If the threshold for adoption for every person is q as we said before.
Then the cascade cannot complete itself i.e., starting from some nodes it cannot sweep the entire network, the cascade cannot complete itself if there exists a cluster in this network of density greater than $1-q$.

Example:

Going outside $L = \cancel{60\%}\,40\%$.
Going Library $L = 60\%$.

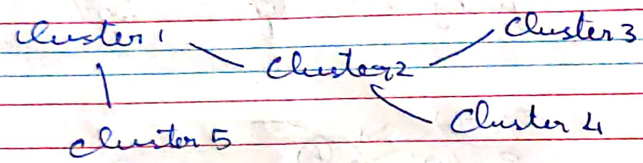


Cluster with threshold greater than $(1-q)$
greater than 0.6

The idea is trying to enter the cluster.
$\geq 60\%$ of the friends of node A are already planned to go to library.
Hence, the threshold cannot be achieved for the ~~b~~ shopping. It will be always less than 40%, Hence you cannot influence the cluster.

Vice versa is ~~only~~ to also true.
If the cascade is not complete, it means that there is a cluster in the network, which has a threshold greater than $1-q$.

Video 8: Knowledge, Threshold and Collective Action

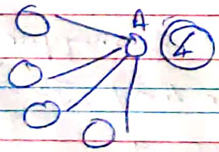


Since everybody in the cluster do not communicate that much with the other cluster, eventhough each cluster is ready to evolve, as they all have the starting trouble.

Collective decision:

It is implemented using Intrinsic threshold.

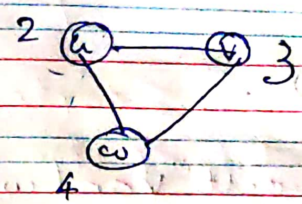
It means



Each people may have several intrinsic threshold. node A assumes that I need atleast 4 people to make a change which is the intrinsic threshold for node A.

Intrinsic threshold varies for each node.

Example 1:

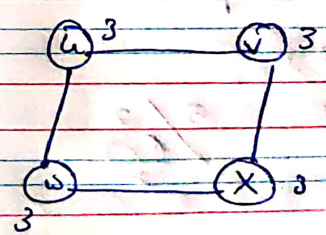


w drops the idea.

v drops the idea.

u checks its neighbours. It sees that both w and v are not participating in the idea. Hence u will also drop the idea of protesting.

Example 2:



u does not know about x's threshold

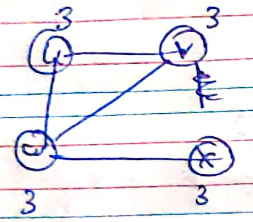
v → w

x → u

w → v

Hence, none of them protest.

Example 3:



u knows about v and w threshold
v knows about w and u threshold
w knows about u & v threshold.

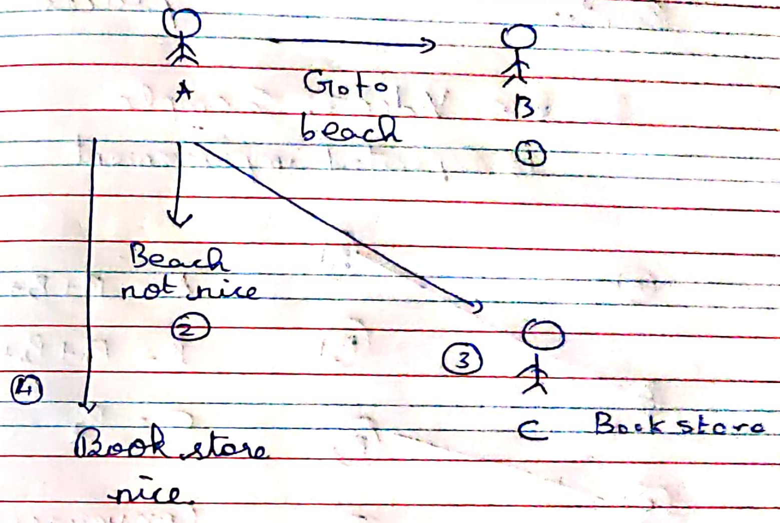
Here, the threshold is considered by including the current node also.

Since u, v, w all require only three nodes and 3 nodes are available for all the three, (u, v, w) goes for a protest.

$\dfrac{6}{\frac{6\times5}{1\times2}}$ $\quad 6/15 \quad$ $5 / \dfrac{\cancel{4}^{2}\times3}{2}$

Week 8: Video 1

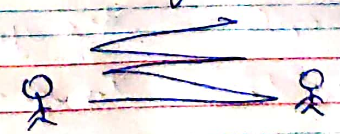
Introduction to hubs & authorities



So, this means that, if the place is good, the person get good points. The person's recommendation will be valued if the person got more points.

Week 8: Video 2:

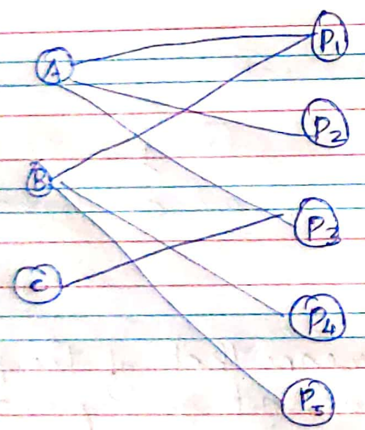
Principle of Repeated Improvement



Repeated Improvement

Better related Lists 'Pointer' ⟶ Better related places 'places'

Week 8: Video 3: Example of Repeated improvement



$A = P_1 + P_2 + P_3$

$B = P_1 + P_4 + P_5$

$C = P_3$

Converges after many iterations

Week 8: Video 4
Hubs & Authorities

Every person has two resources?

Are good people pointing at you?

Are you pointing to good people?

How to achieve this? ⟹ Repeated improvement.

9,10,11,12 week pending