

# NETWORK VISUALIZATION & SOCIAL NETWORK ANALYSIS

Team members: Anh Dang, Cicely Lambright

Advisor: Prof. Fred Annexstein

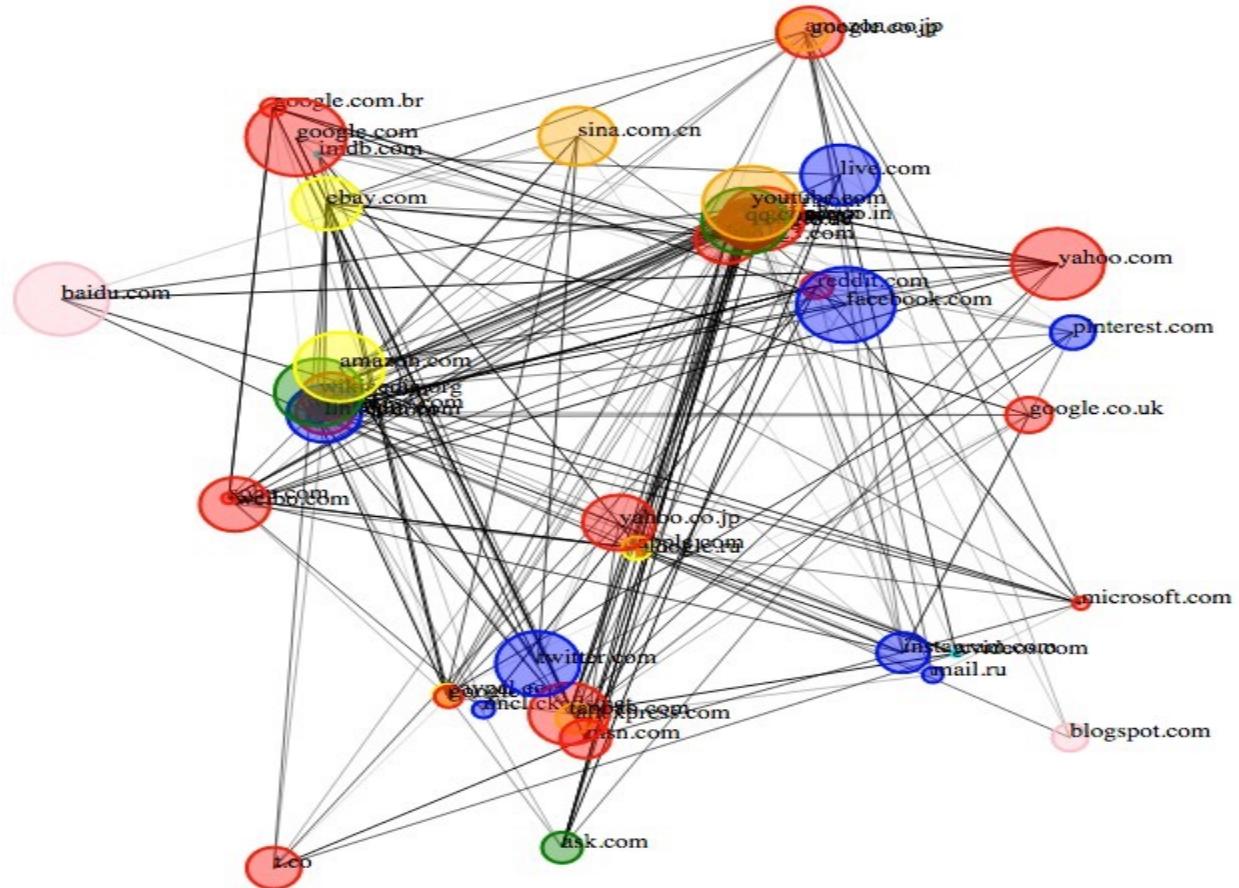
# NETWORK VISUALIZATION

# BACKGROUND

- Network visualization strategies (Interaction Design Foundation, 2016)
  - Change the layout for the graphical design to improve the readability of the design
  - Create an approximation of the overall structure but reduce the complexity so that it may be easier to comprehend
  - Implement interactivity so that the user can manipulate the representation to match their needs

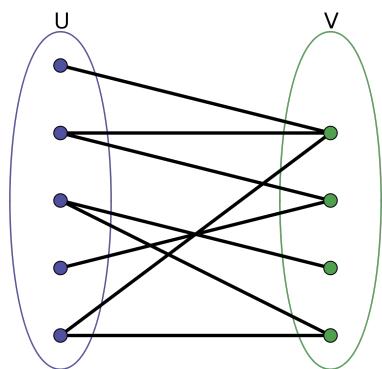
## BACKGROUND

- Our focus: changing layout to reduce edge crossing
  - Common approaches:
    - Force direction
    - Spring embedder

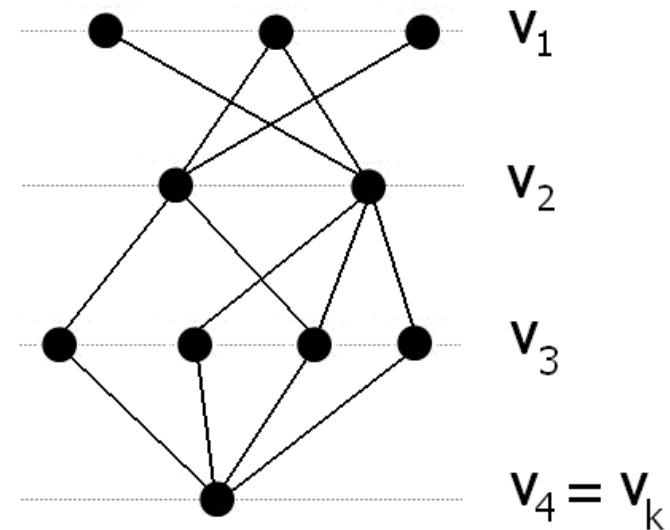


# BACKGROUND

- Advanced methodologies in crossing minimization
  - Find the maximum planar subgraph and add remaining edges one-by-one
  - Transform into a k-layer graph of bipartite and reduce crossing of each layer



Bipartite graph



'Proper' Layered graph

# BACKGROUND

- Obstacles in the field:
  - Finding minimum crossing number is NP-hard
  - No existing solution that is both practical and optimal
  - Previous work is sub-optimal: proving their effectiveness by upper bounds

# GOALS

- Develop a method from existing techniques that finds a graph layout with lower number of crossing: layering methodology

# INTELLECTUAL MERITS

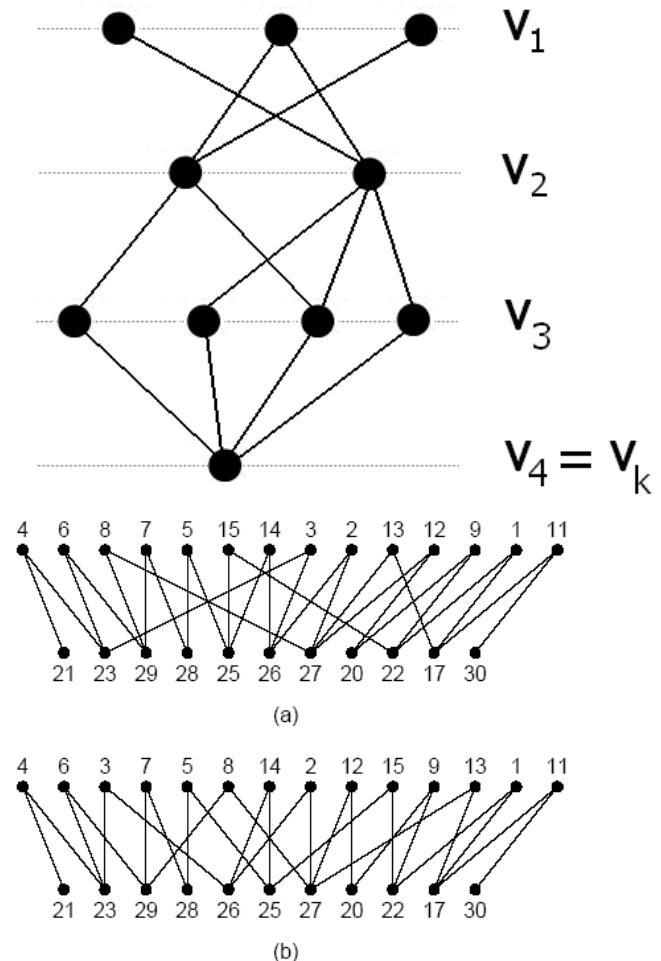
- The proposed method is straightforward and easy to implement
- Suboptimal
- Ensure ‘reasonable’ runtime
- Proven to provide less crossings than default plotting functions and a chosen base method: limited experiments

# BROADER IMPACTS

- Application: network design, organizational grouping, industrial delivery system
- Provide a ‘better’ but simpler solution without reading papers and writing own codes

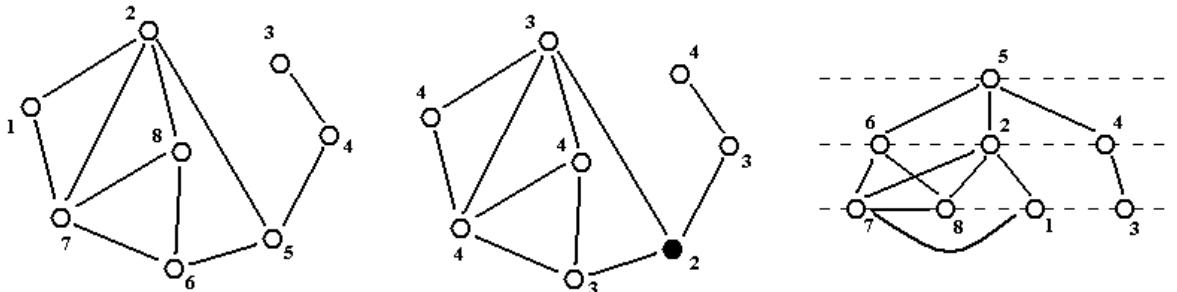
# FRAMEWORK

- Sugiyama et al. (1979)
- Step 1: Layering
  - Convert the entire graph into layers
  - Remove flat edges
  - Replace long edges with short edges using dummy edges
- Step 2: Crossing reduction
  - Choose a heuristic to solve one-sided two-layer crossing minimization problem
  - Layer-by-layer sweeping

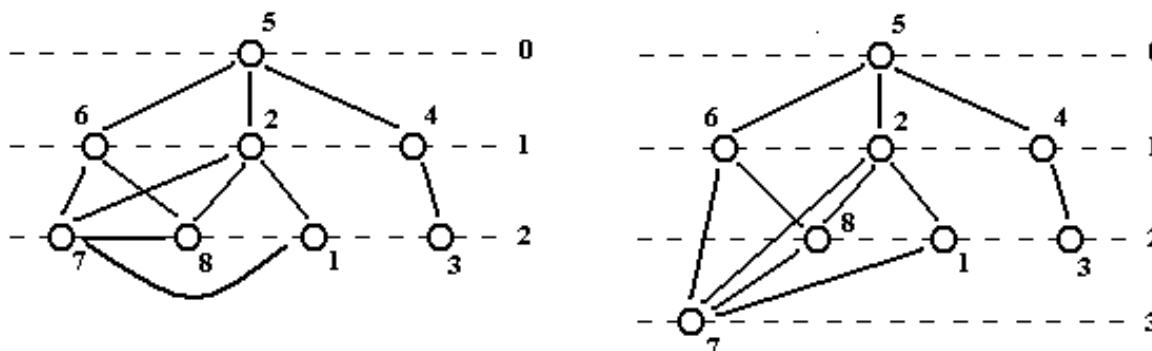


# DESIGN SPECIFICATIONS

- CROSSING MINIMIZATION
  - Adopted methodology (Sugiyama method) - Overview
    - Step 1: transform into k-layer graph



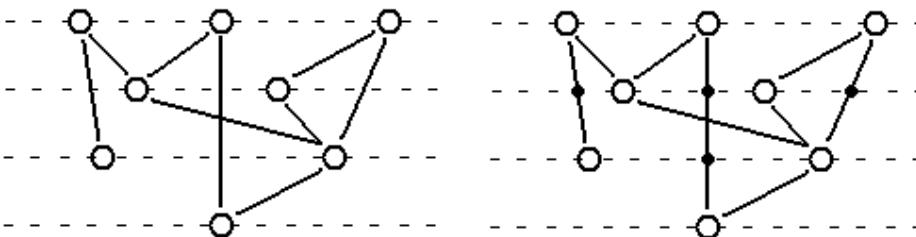
- Step 2: remove flat edges



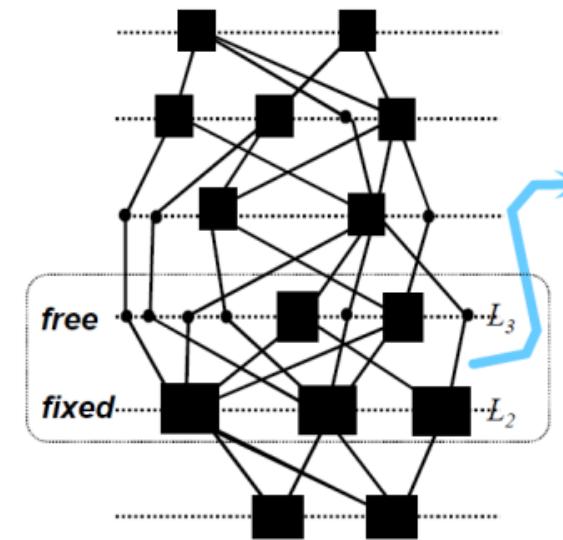
(Kieft, 2006, Hong, 2018)

# DESIGN SPECIFICATIONS

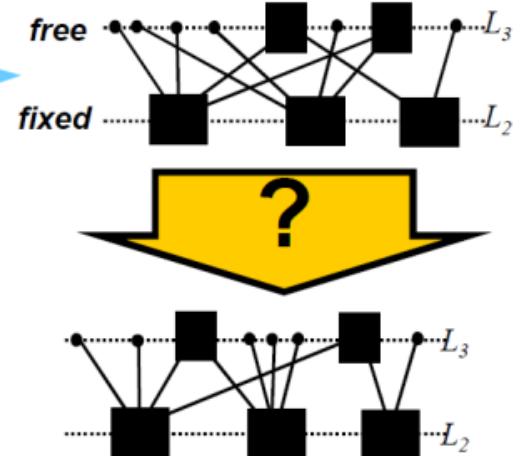
- CROSSING MINIMIZATION
  - Adopted methodology - Overview
  - Step 3: remove improper edges



Step 4: Layer-by-layer sweeping



The difficult part is to  
re-arrange the free layer



(Kieft, 2006, Hong, 2018)

# DESIGN SPECIFICATIONS

- CROSSING MINIMIZATION
  - Adopted methodology - Overview
    - Step 4: Layer-by-layer sweeping
    - Heuristic to solve one-sided two-layered problems: Barycenter (Sugiyama et al., 1981)
    - Best performer in terms of efficiency and speed (Jünger M., Mutzel P., 1996)
    - x-coordinate of each vertex  $u$  in  $L_2$  is chosen as the barycenter(average) of the x-coordinates of its neighbors

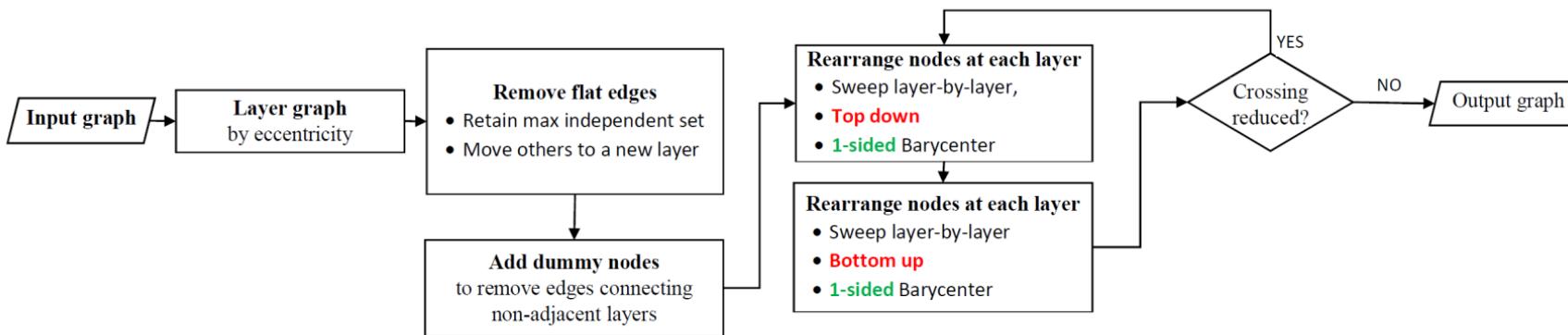
$$\text{bary}(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} \pi_1(v)$$

- If two vertices have the same barycenter then order them arbitrarily

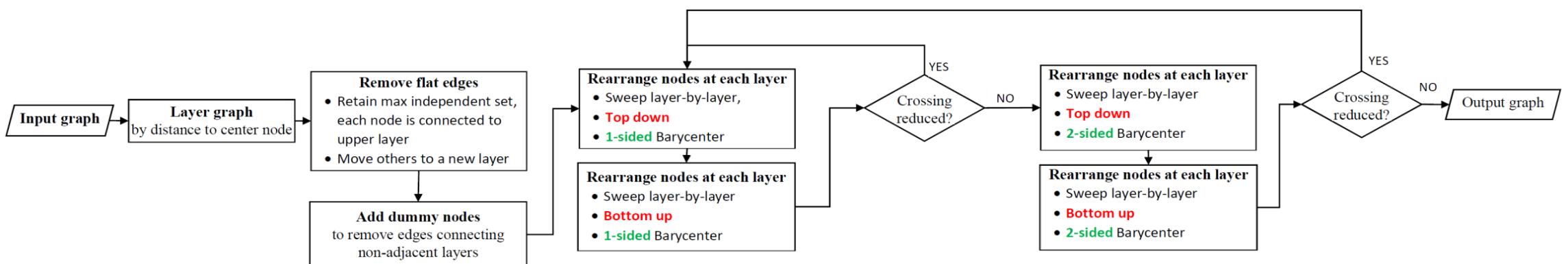
# DESIGN SPECIFICATIONS

- CROSSING MINIMIZATION

- Base method for comparison (Kieft, 2016, Hong, 2018)

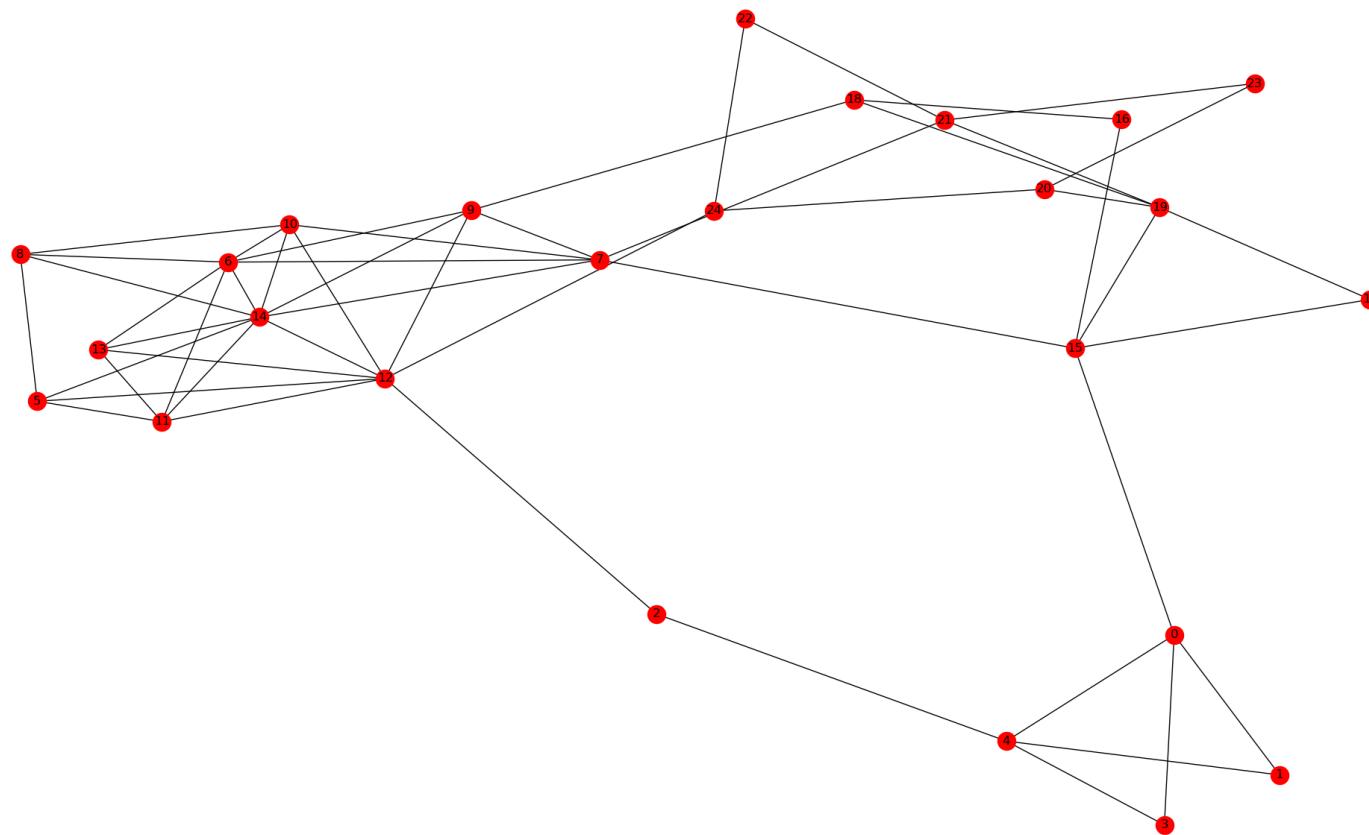


- Proposed method



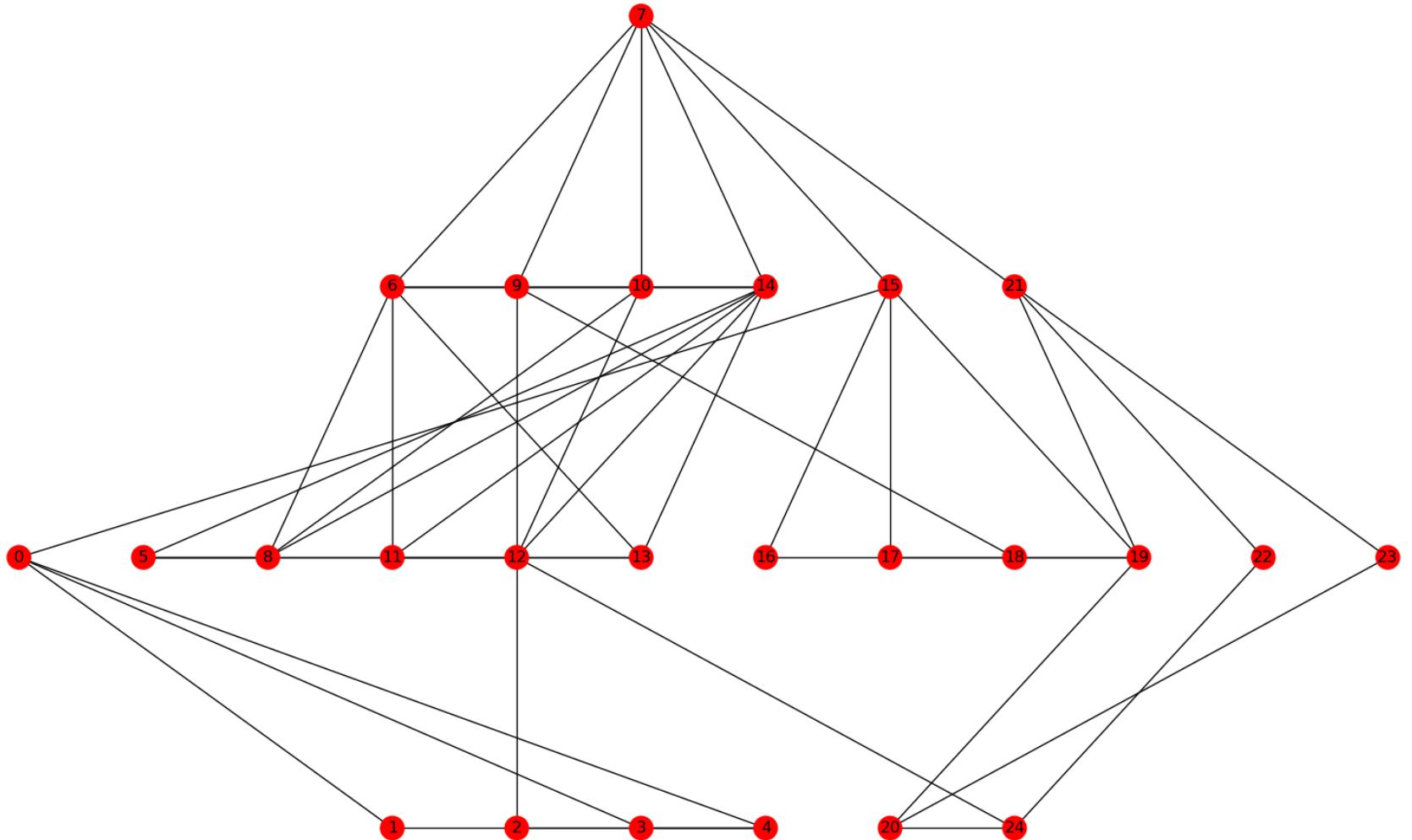
# DEMONSTRATION

- Graph sample – default plot function from Network X



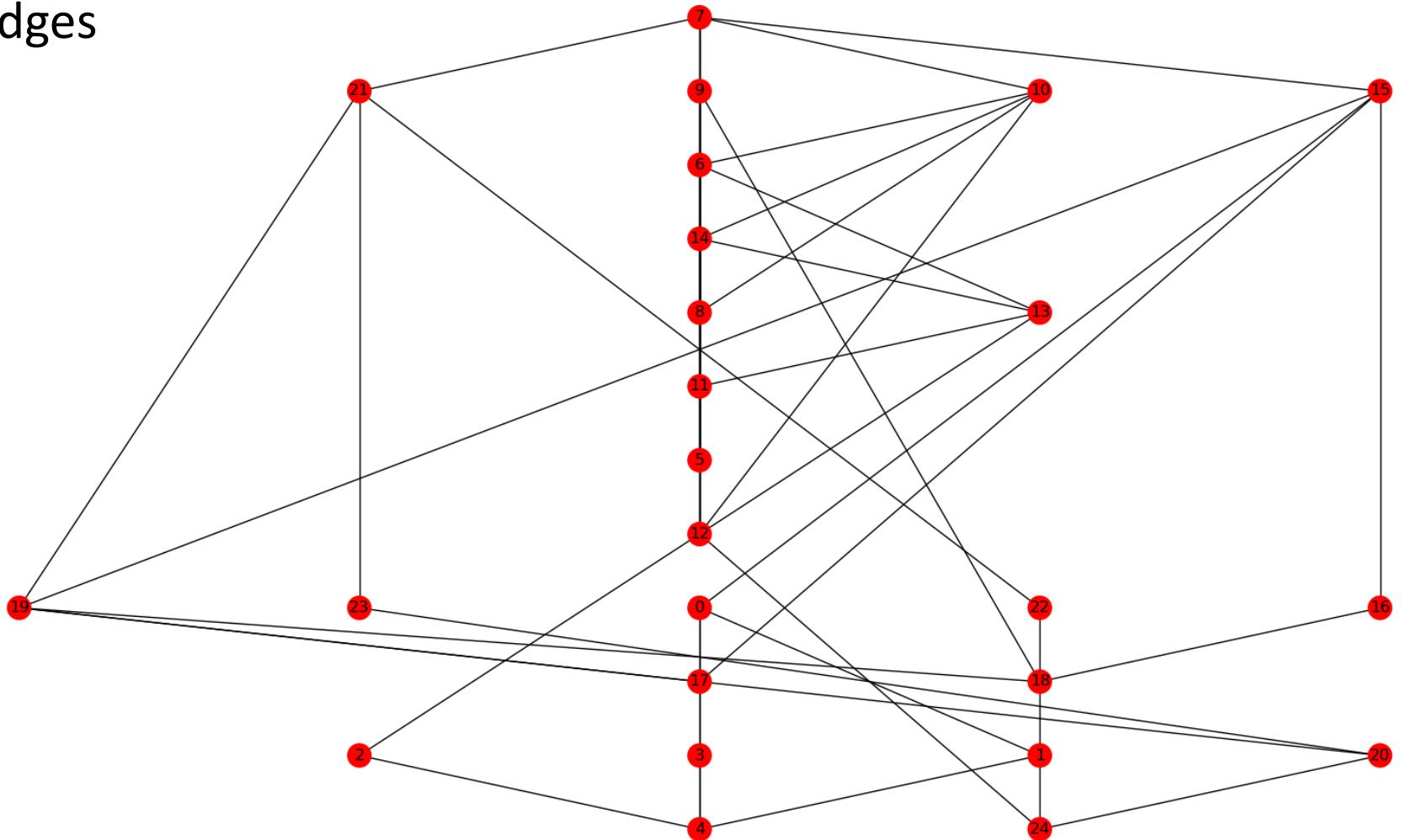
# DEMONSTRATION

- Step 1: layering



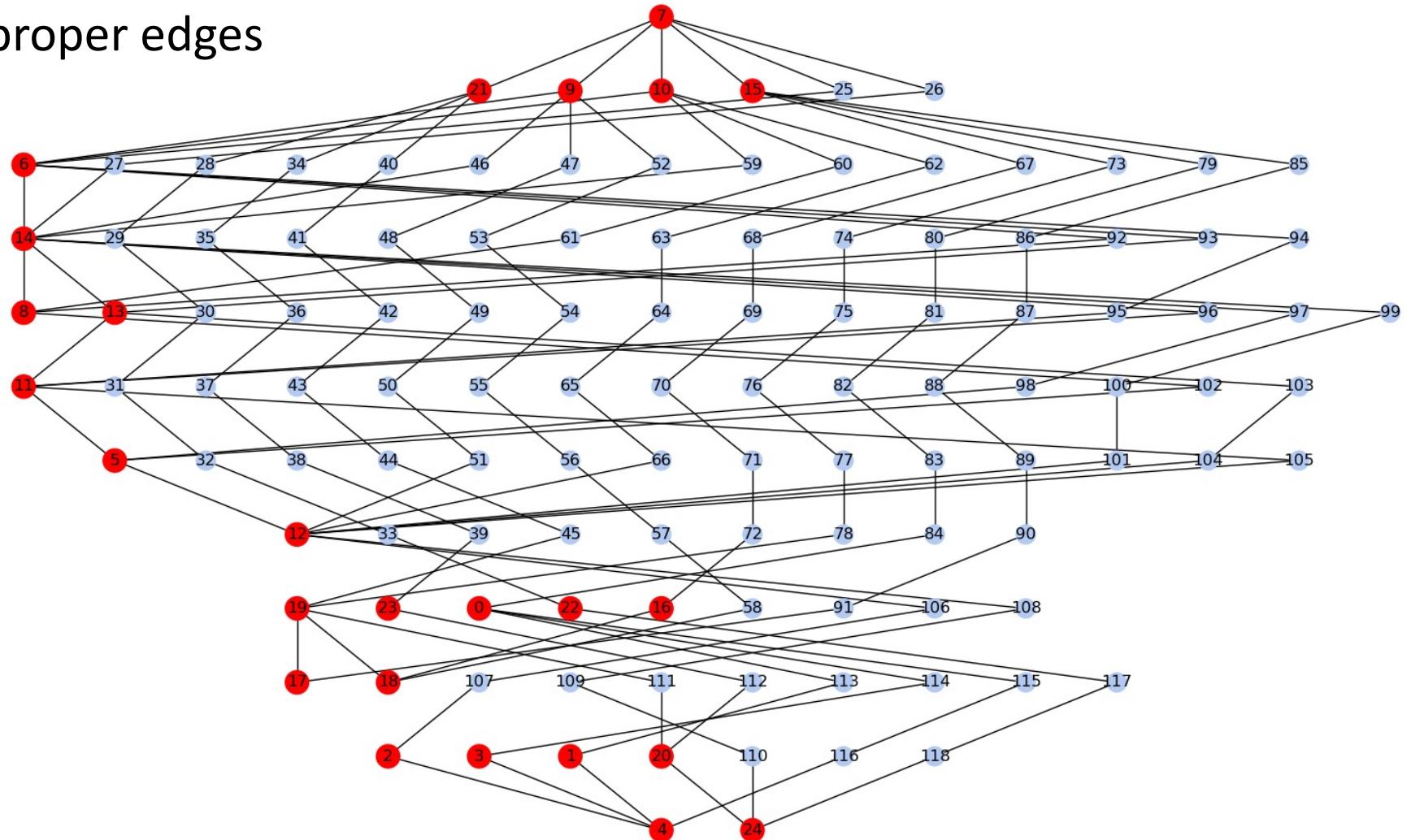
# DEMONSTRATION

- Step 2: removing flat edges



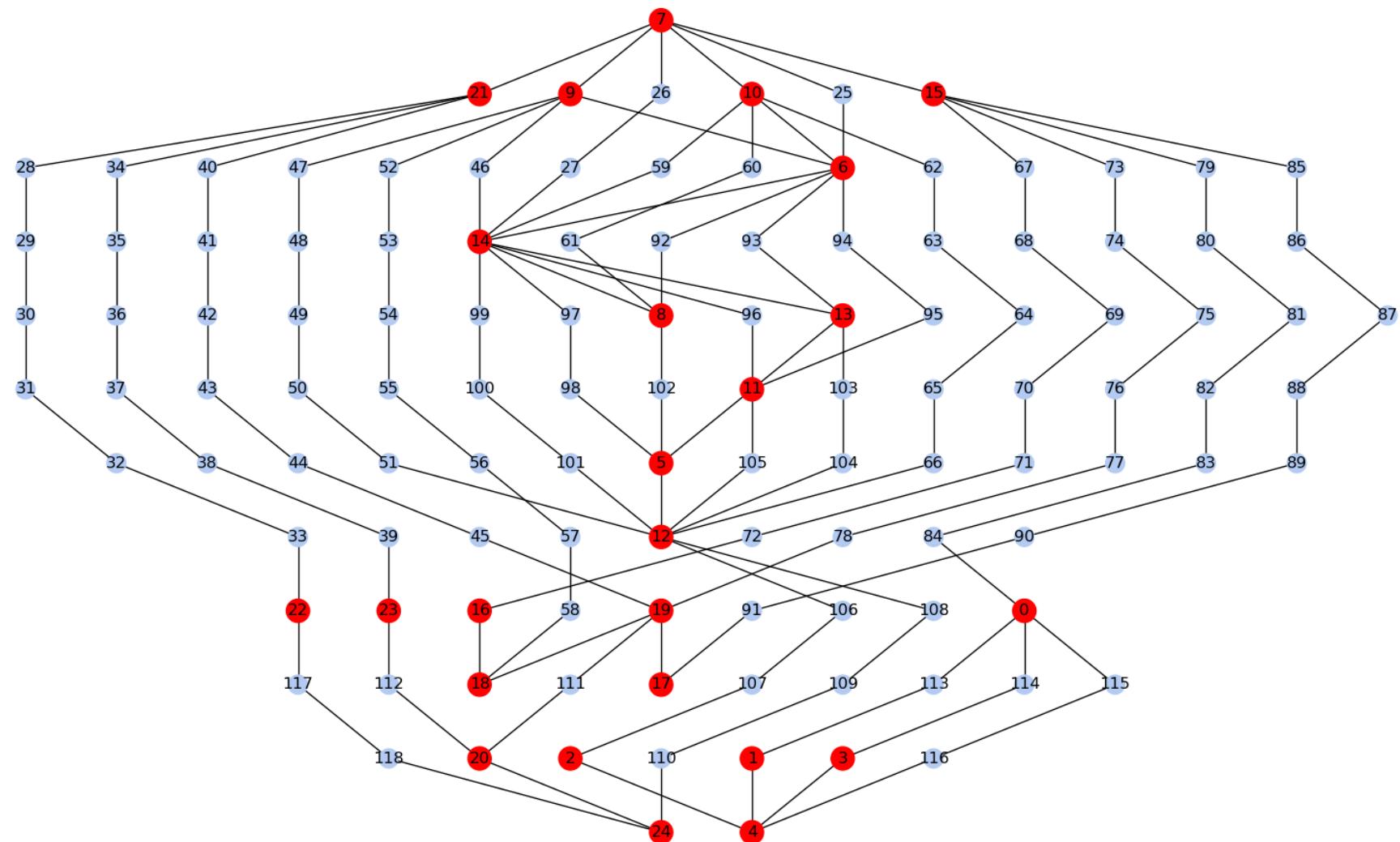
# DEMONSTRATION

- Step 3: removing improper edges



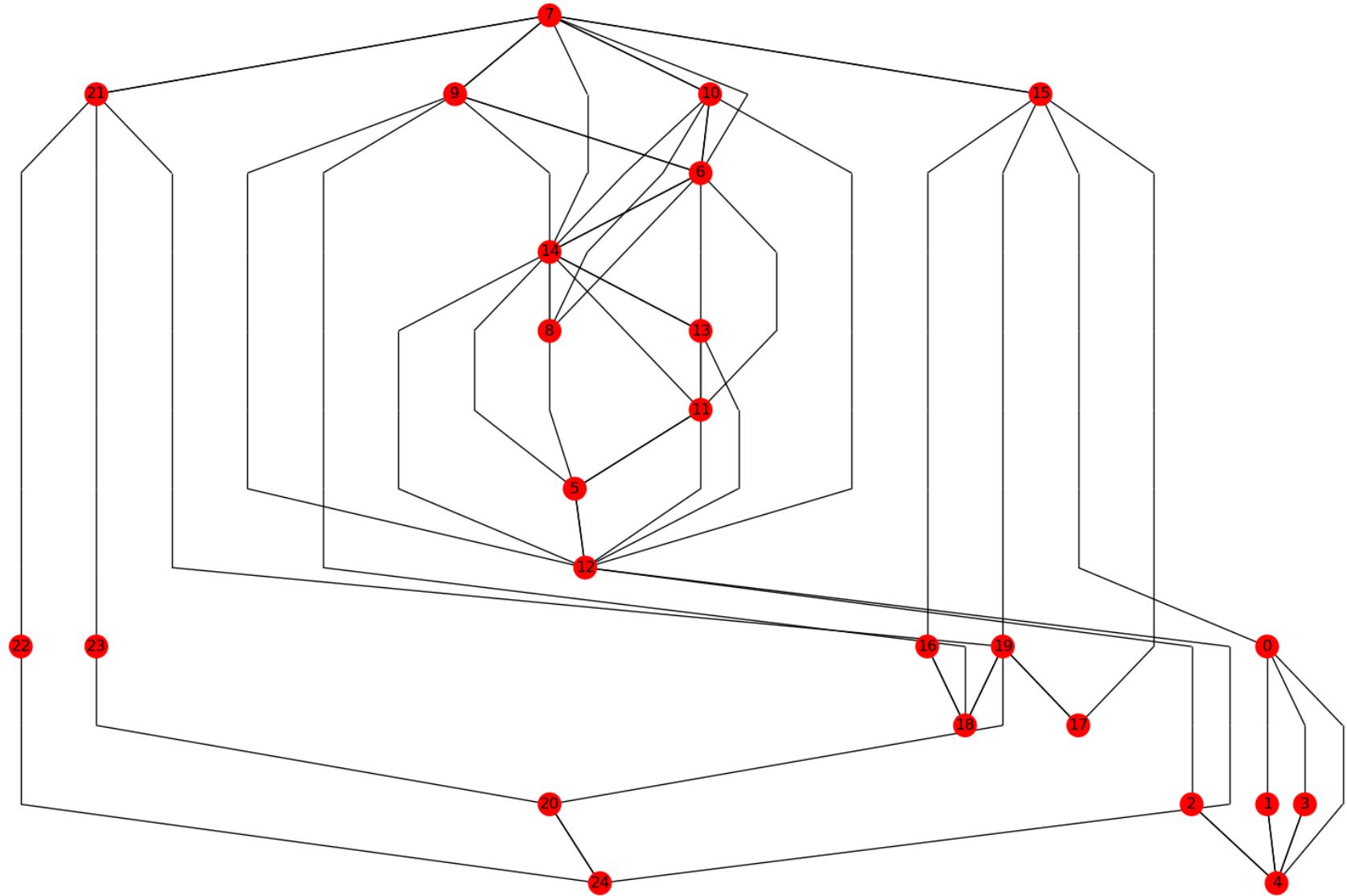
# DEMONSTRATION

- Step 4: sweeping



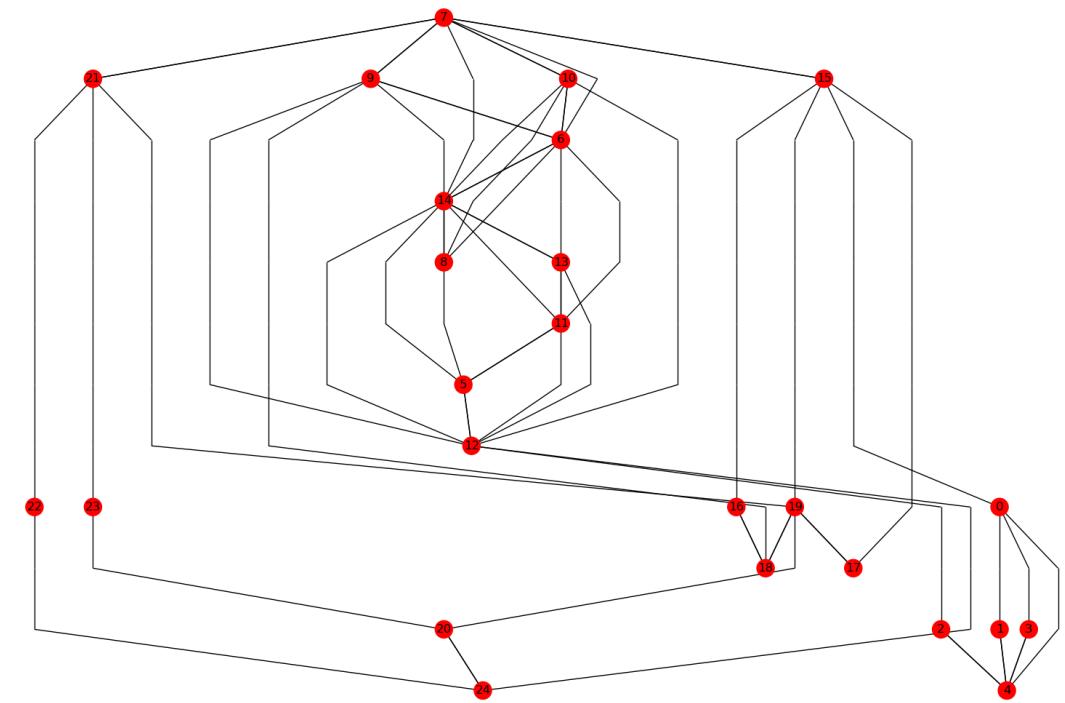
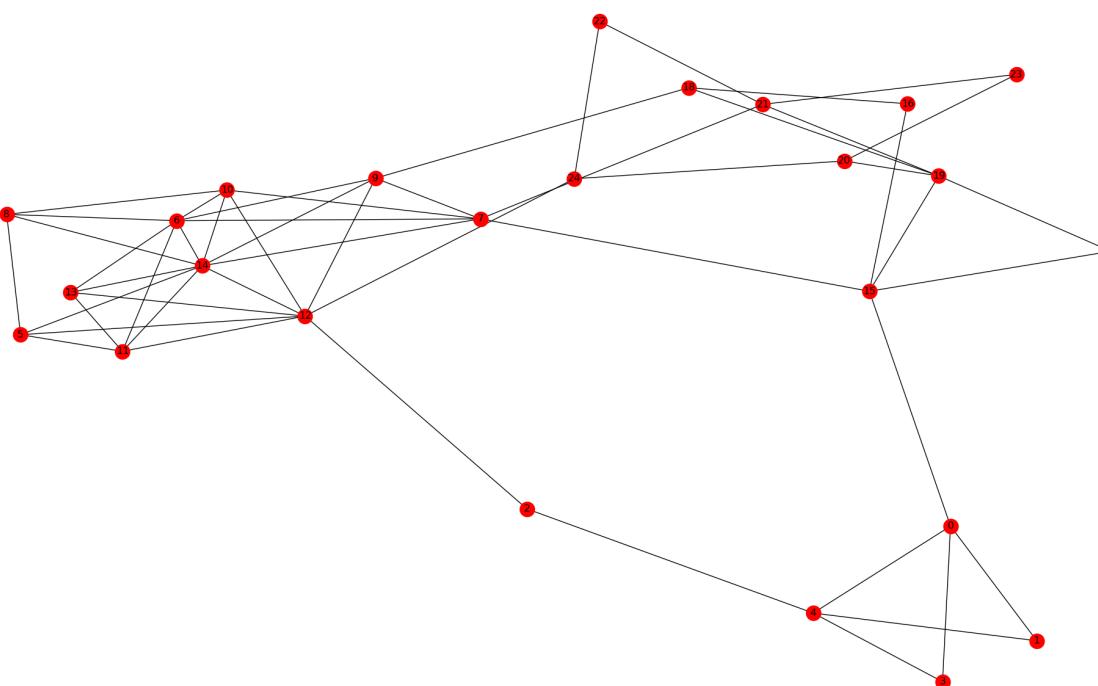
# DEMONSTRATION

- Step 5: postprocessing



# DEMONSTRATION

- CROSSING MINIMIZATION
  - Comparison



# RESULTS

GRAPH		NUMBER OF CROSSING		
# of nodes	# of edges	NetworkX default plot	Base method	Proposed method
25	51	No algorithm for crossing calculation	33	27
25	161		1,592	<b>1,363</b>
40	127		616	<b>431</b>
40	393		10,230	<b>9,614</b>
80	491		12,344	<b>8,043</b>
80	1554		175,311	<b>167,098</b>

# SOCIAL NETWORK ANALYSIS

# ALGORITHM OVERVIEW

- Collect user keywords/usernames/hashtags/subreddits
- Collect data from choice of social network (reddit, twitter)
  - Make sure there's a limit for the data
- Parse the data
- Calculate importance for each node
- Remove less-important nodes from the graph
  - Current model – approx. 90% removal
- Output
  - Graph file
  - Visual data file

# NODE IMPORTANCE

- Metadata Components
  - Twitter
    - Which batch of tweets (original keywords or secondary)
    - Favorites
    - Connections
    - Retweets
    - Followers
    - Friends
  - Reddit
    - Comment and post karma
    - Reddit Gold
- Programmatical Components
  - Closeness Centrality
  - Betweenness Centrality
  - Degree Centrality
  - Eigenvector centrality
  - PageRank

# NORMALIZING NODE TYPES

Different node types (hashtag vs user) may be unfairly skewed in importance based purely on the fact that they occur in different instances, to remove such skew, we used the formula below:

*Assuming we have two lists:*

$$\begin{bmatrix} a_0 \dots a_n \end{bmatrix}$$
$$\begin{bmatrix} b_0 \dots b_n \end{bmatrix}$$

*If there is an element  $x$  in list A  
that we want to convert to a  
value  $y$  in list B in a similar  
position to A in relation to the  
list, but a different value, we can  
use the formula:*

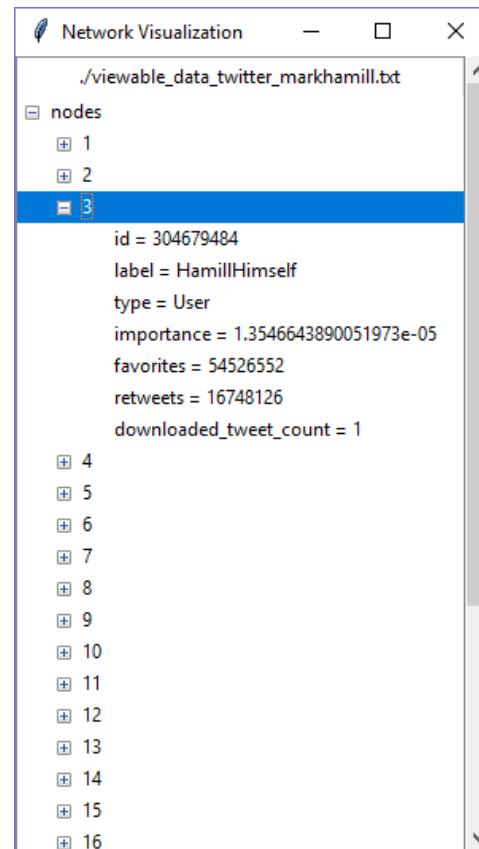
$$y = (x * \frac{b_n - b_0}{a_n - a_0}) + (b_0 - a_0)$$

*Where if     $b_n - b_0 = 0$   
or                 $a_n - a_0 = 0$*

*We use 1 instead of 0 for  
whichever values are = 0*

# NODE INFORMATION VIEWING

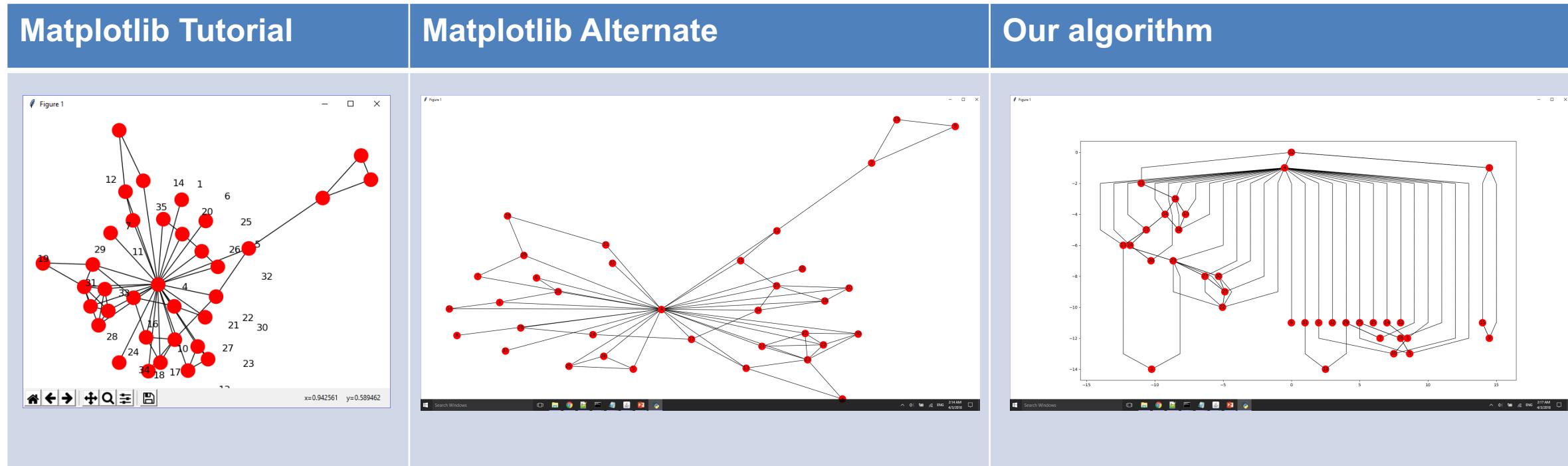
Created program for easy node information viewing



# RESULTS

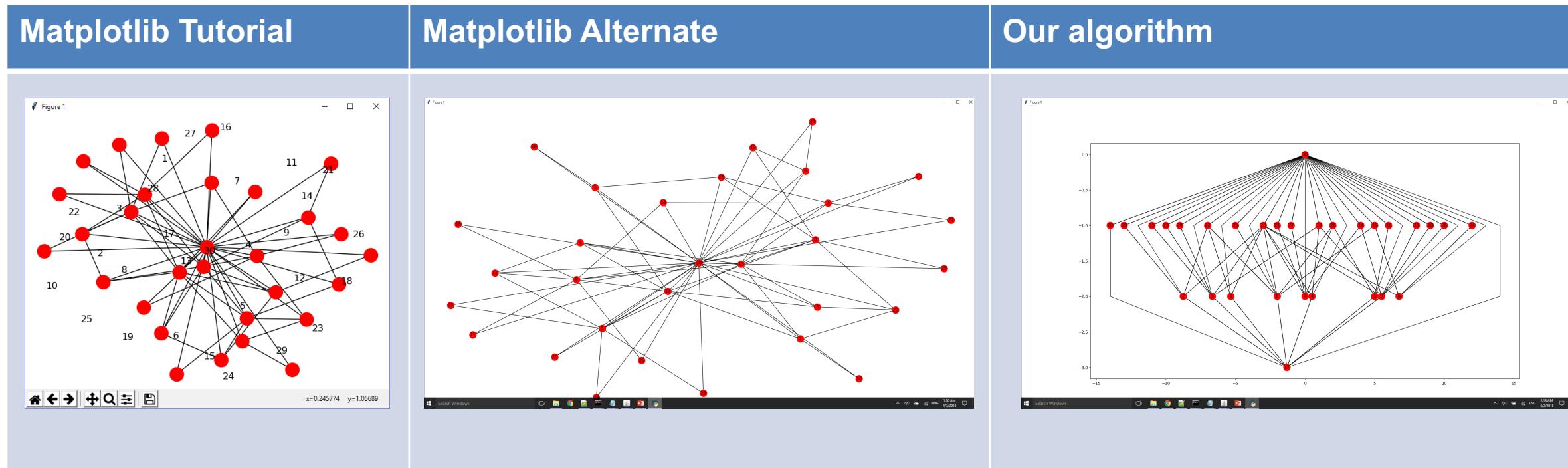
# EXAMPLE: @UOFCINCY

Platform	Type	Target	Number Nodes
Twitter	User	@uofcincy	35



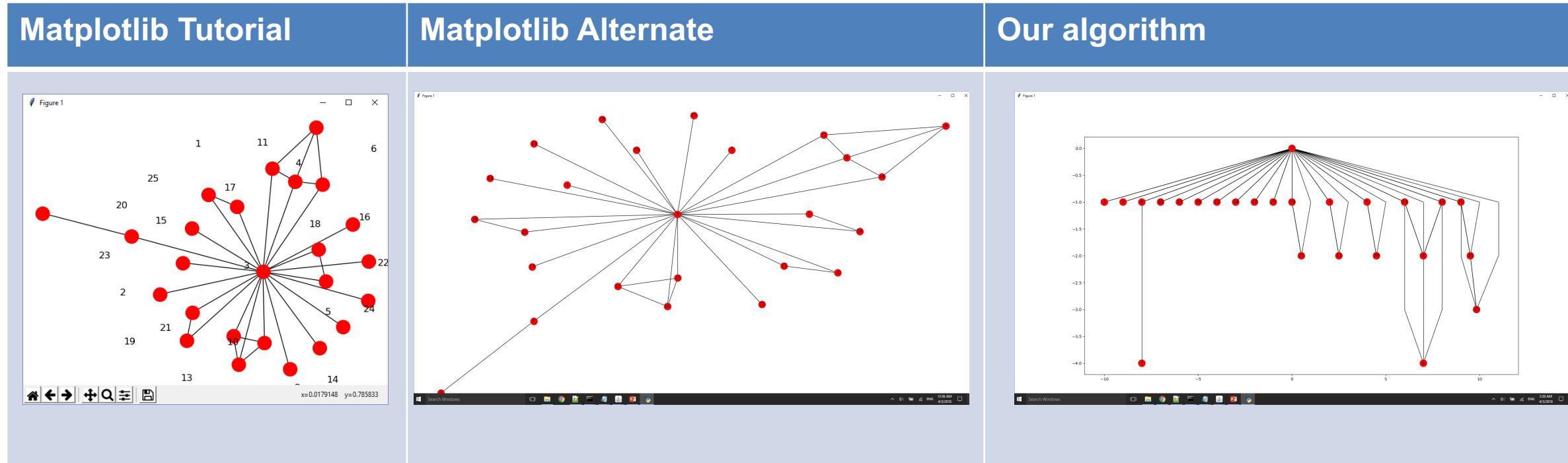
# EXAMPLE: @R/DANKMEMES

Platform	Type	Target	Number Nodes
Reddit	Subreddit	dankmemes	30



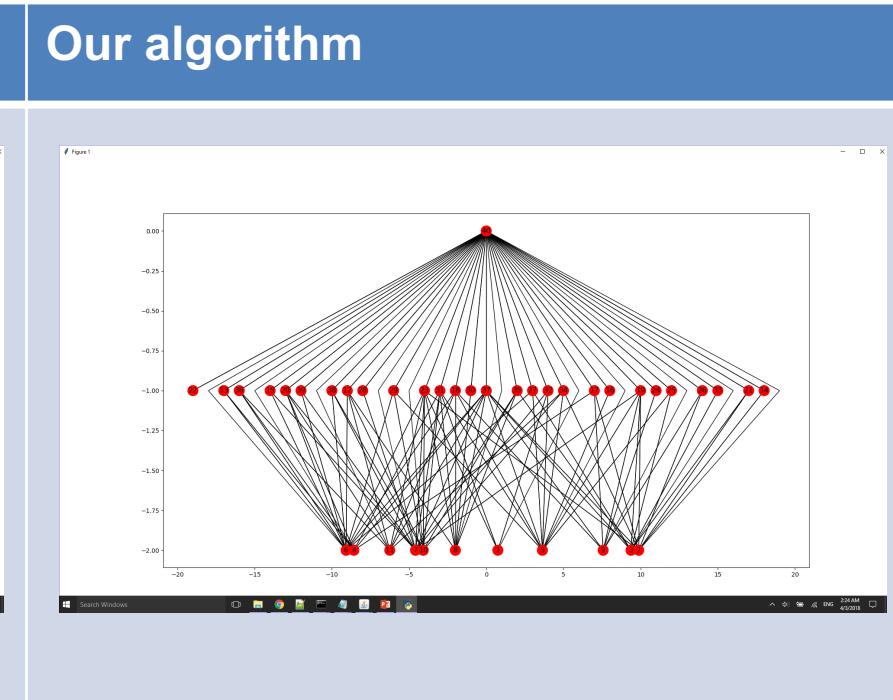
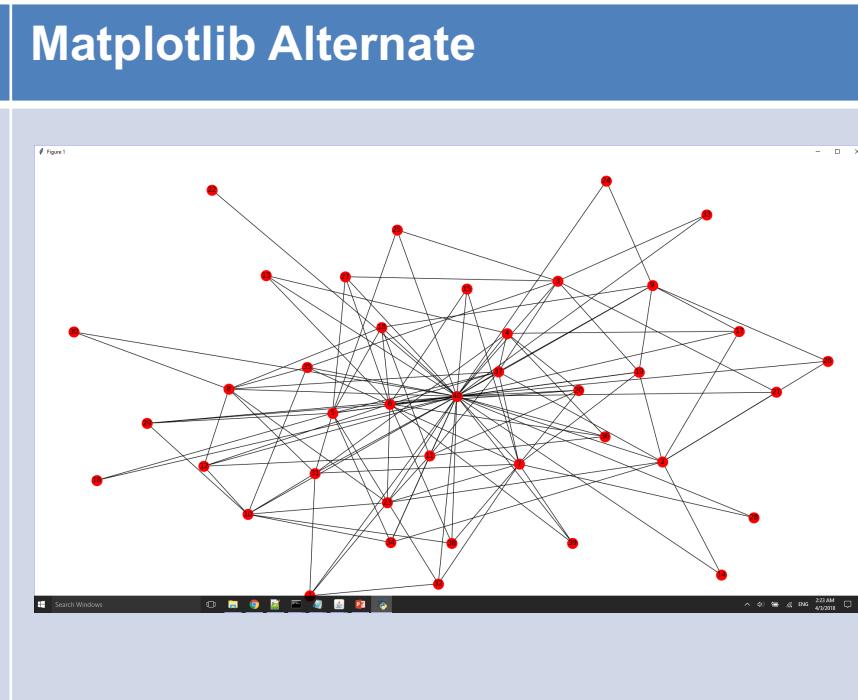
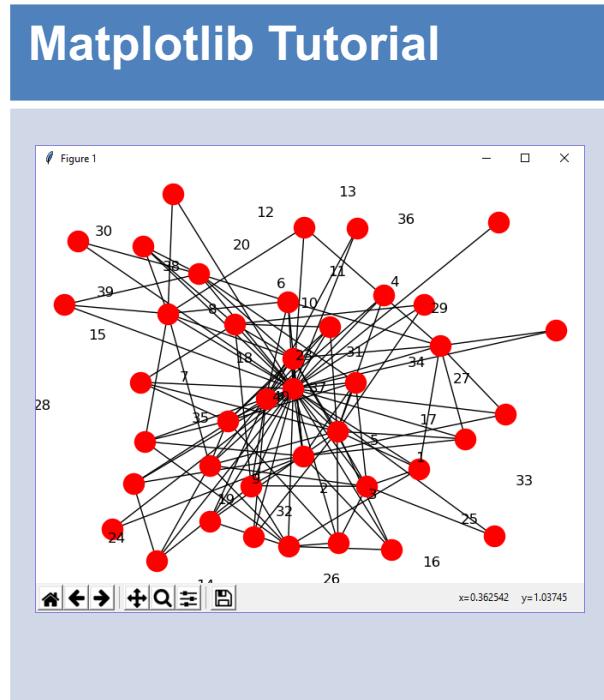
# EXAMPLE: @HAMILLHIMSELF

Platform	Type	Target	Number Nodes
Twitter	User	@HamillHimself	25



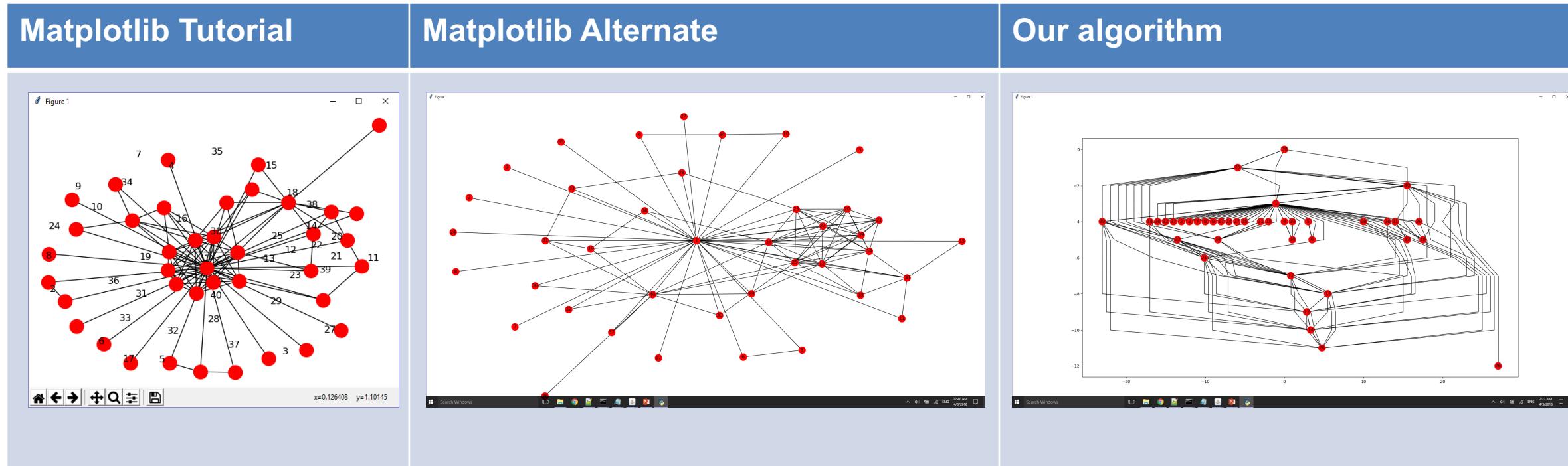
# EXAMPLE: R/PROGRAMMERHUMOR

Platform	Type	Target	Number Nodes
Reddit	Subreddit	ProgrammerHumor	40



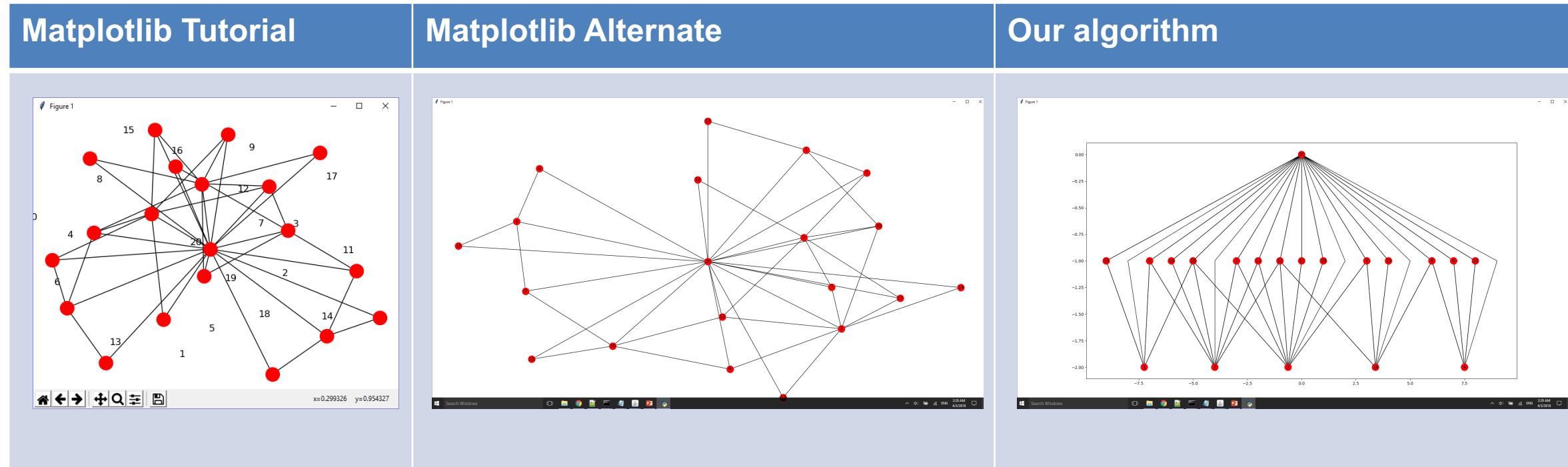
# EXAMPLE: #UNDERTALE

Platform	Type	Target	Number Nodes
Twitter	Hashtag	#Undertale	40



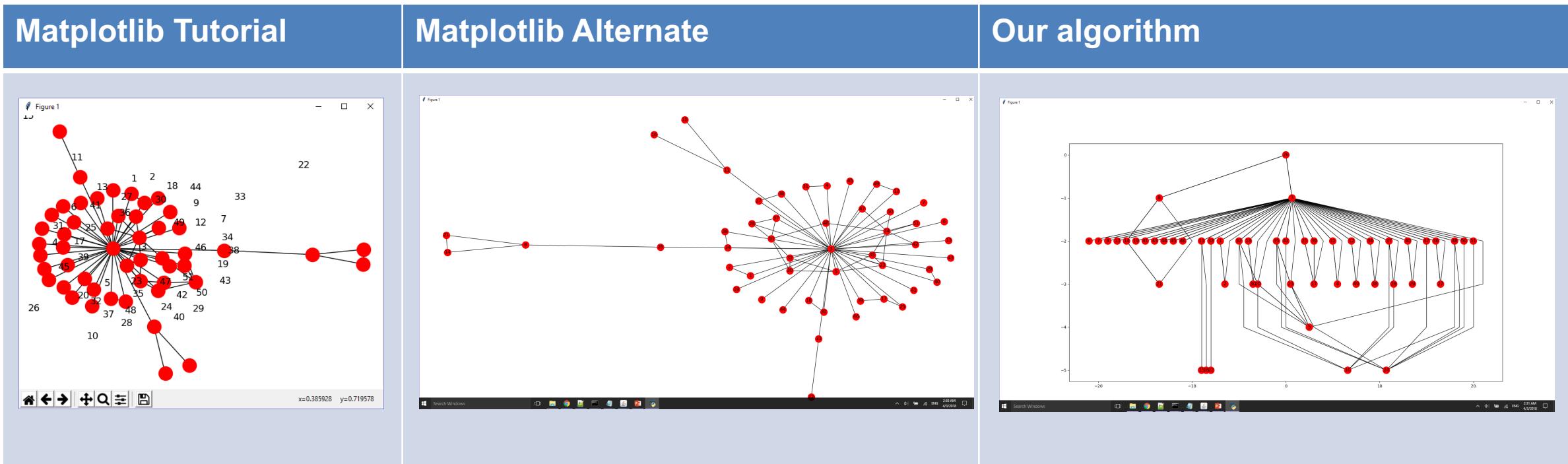
# EXAMPLE: R/BREADSTAPLEDTOTREES

Platform	Type	Target	Number Nodes
Reddit	Subreddit	BreadStapledToTrees	20



# EXAMPLE: @REALDONALDTRUMP

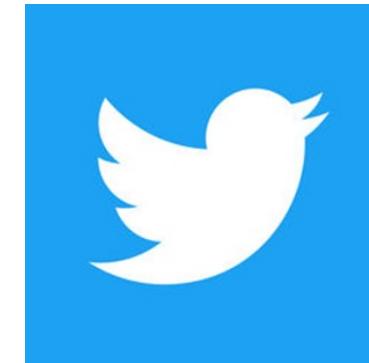
Platform	Type	Target	Number Nodes
Twitter	User	@realdonaldtrump	50



# Technologies



**NetworkX**



**NLTK**

