# CS 6150: Advanced Algorithms Project Update 1
# Maximizing the influence of a social network

Aakanksha Saha and Shuvrajit Mukherjee

School of Computing, University of Utah

November 28, 2017

***Task Implemented 1:***

- Estimated the spread of influence using the Greedy approach mentioned in the paper.
  ***Decision Choices:***
  - We have implemented the Independent Cascade model to calculate the spread.
  - We have used the following two data set to verify our result:
    1. Slashdot Social Network(82168 Nodes, 948464 Edges, Directed)
    2. Social Circle: Facebook(4039 Nodes, 88234 Edges, Undirected)
  - Random seed set to 57
  - Any node once influenced by a node once is not double counted in later steps.
  - The problem being NP-Hard, we were not able to compare it with the optimal solution, So we have compared our results with randomly selected $k$ nodes

  ***Experiment Results:***

  - Slashdot Data set
    - k = 10
    - Spread using greedy approach: 6567 nodes
    - Spread while k nodes are selected randomly: 90

  - Social Circle: Facebook
    - k = 4
    - Spread using greedy approach: 2198 nodes
    - Spread while k nodes are selected randomly: 341

***Task Implemented 2:***

- Estimated the spread of influence using the Highest degree heuristic.
  ***Decision Choices:***
  - We have implemented the Independent Cascade model and the Weighted Cascade model to calculate the spread.
  - We have used the following dataset to verify our result:
    1. Slashdot Social Network(82168 Nodes, 948464 Edges, Directed)
  - Used networkx module in python to read the graphs and traverse it, used generators to get a space efficient method for large data processing.

  ***Experiment Results:***

  - Slashdot Data set
    -Seed value of k = 10
    - Independent cascade model was able to influence $\approx$ 24k nodes. Running Time $\approx 5 seconds$
    - The weighted cascade model on the other hand could only influence 300 nodes.

***Proposed Heuristics and Future work***

- As an extension to highest degree heuristic we would like to implement the degree discount heuristic: `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/kdd09_influence-1.pdf`. The general idea is as follows; let v be a neighbor of vertex u. If u has been selected as a seed, then when selecting v as a new seed based on its degree, we should not count the edge vu towards its degree. Thus, we should discount v's degree by one due to the presence of u in the seed set, and we do the same discount on vs degree for every neighbor of v that is already in the seed set.

- Also, one of the rough heuristics could be to cover the graph breadth wise and greedily involve the node which is farthest from the given visited node. Thus, we can try to maximize the total distance in the chosen set of seed values or k nodes, thereby covering the graph holistically.

- The following proposed heuristic is based on the following assumption:
  If the graph is sufficiently large the most influential nodes are uniformly distributed across the graph.
  We compute the sphere of a influential node. So, we start with a void set and append the first node with highest influence independently. The next highest influential node will not lie in the sphere of the previous node and after appending this node we will have two spheres, which may have some overlap. We go on like this to add a total of k nodes.