PSAIIM: A PARALLEL SOCIAL BEHAVIOR-BASED ALGORITHM FOR IDENTIFYING INFLUENTIAL USERS IN SOCIAL NETWORKS

Presented by
22i-1242 Anoosha Ali
22i-1046 Afsah Areeb
22k-4360 Saleha Irum

INTRODUCTION

- Why do we need to find **influence user**?
- In order to spread a message quickly through social media we need **influence user**
- What is **Influence Maximization (IM)**?
- The problem of **influence maximization** can be defined as identification of a set of k network users that maximizes the number of users receiving messages

WHY PSAIIM IS BETTER?

Older

1. Ignored semantics
2. Slow and unscalable for large networks
3. Treated all actions equally
4. Most parallel models ignore semantics

PSAIIM

1. Adds semantics:
   - user interests + interaction behavior
2. Uses parallelism for faster execution
3. Weighs interactions
4. First to combine semantics with parallel processing

BASIC DEFINITIONS

1. Community
2. Strongly Connected Community - SCC
3. Connected Acyclic Community - CAC
4. Directed Acyclic Graph - DAG
5. Direct neighbor of a node

BASIC DEFINITIONS

6 Border of a node

7 Semantics of the network

8 Vector characteristic of the user

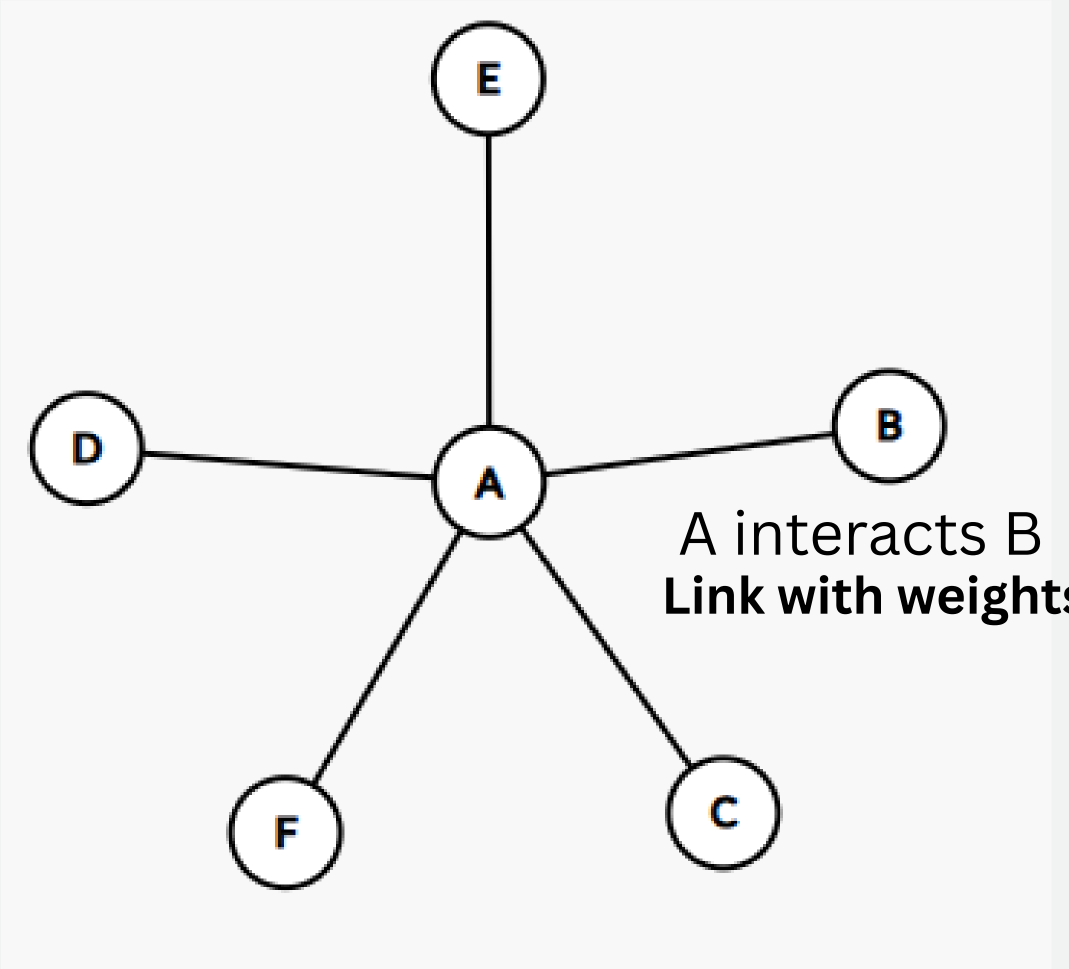
9 Active node

10 Area of Influence

BASIC STRUCTURE OF GRAPH

Vector characteristic of the user

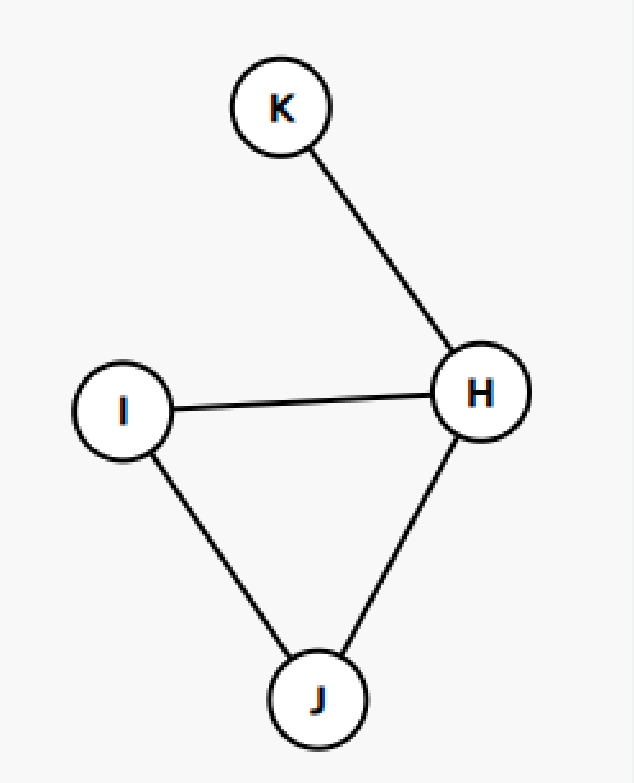
```
User A: {
  interests: ["sports", "memes", "tech"]
}
```



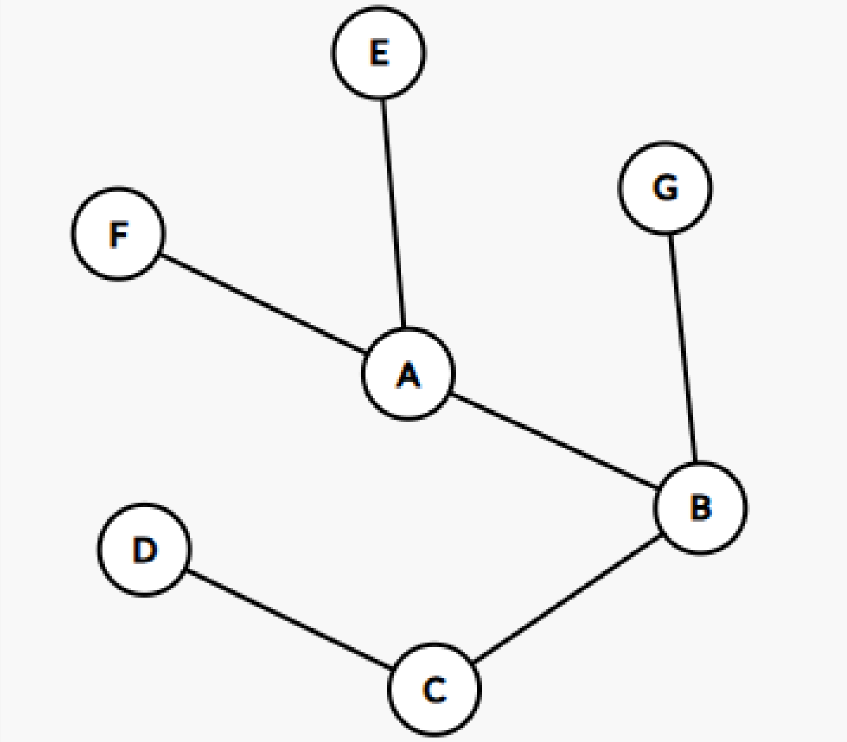
Border of a node

Direct Neighbor of A
B, C, D, E, F

COMMUNITIES

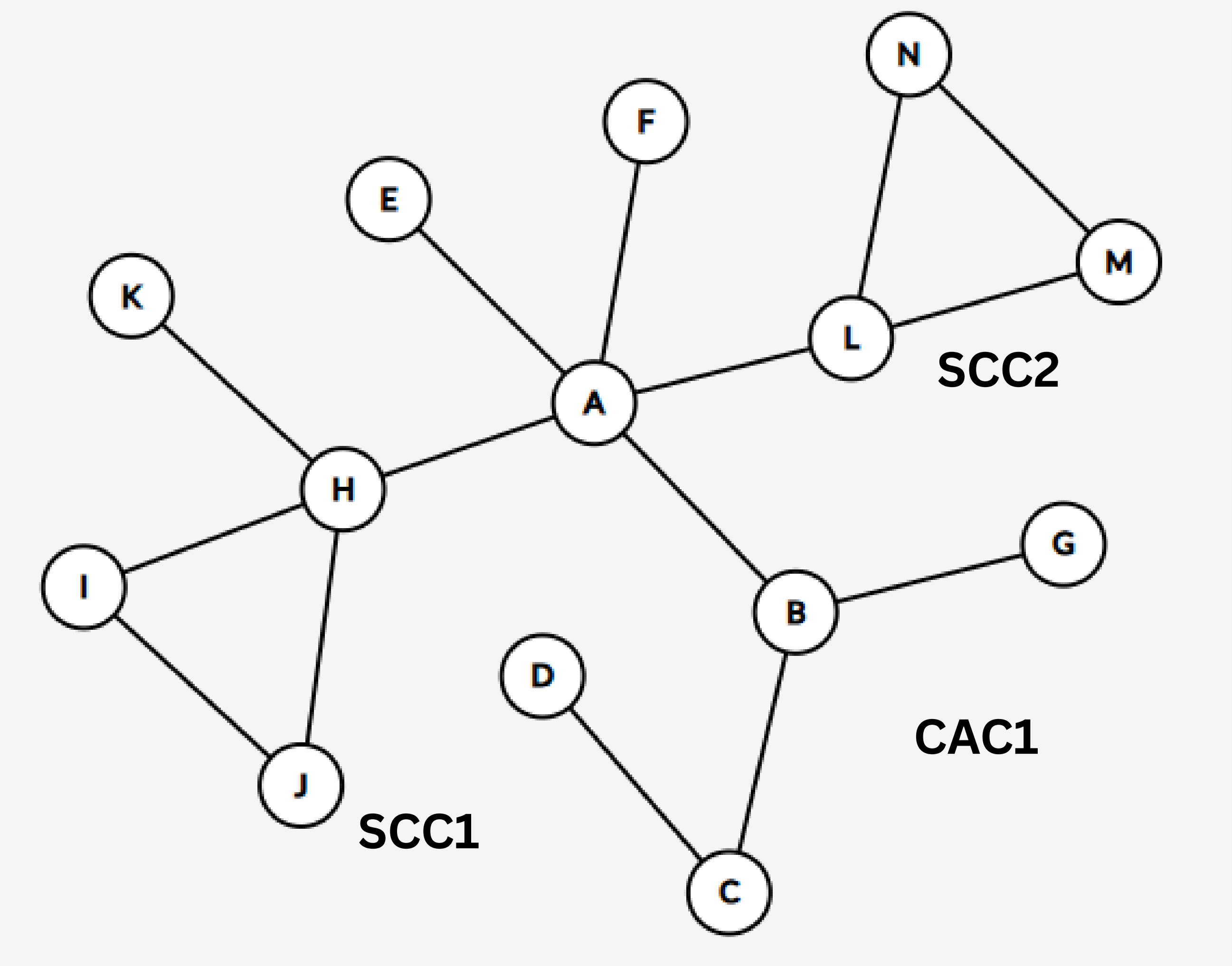


Strongly Connected Community
Every **Node** can reach every other **Node**



Connected Acyclic Community

DIRECTED ACYCLIC GRAPH



TWO MAIN PHASES OF PSAIIM

Phase I

Influence Power Calculation

Phase II

Influential Node Selection

PHASE I - INFLUENCE POWER CALCULATION

- Combine user behavior + interests to understand influence.
- Use PageRank to assign influence scores to each user.
- But PageRank is slow for big graphs!

PROBLEMS WITH PAGERANK

- To calculate a node's score, you need other node scores.
- Creates dependency: A needs B, B needs C, etc.
- Not easy to parallelize.

SOLUTION - GRAPH PARTIONING

- Break graph into smaller parts:
  - SCC (Strongly Connected Components)
  - CAC (Connected Acyclic Components)
- These groups are easier to compute separately.

ASSIGN LEVELS WITH DFS

- Use Depth First Search to give levels to each group:
  - Level 0: No dependency
  - Level 1: Depends on level 0
  - Level 2: Depends on level 1, and so on...
- Compute PageRank level by level in parallel.

SEED CANDIDATES SELECTION

After computing influence scores using PageRank:

- For each node v, define its Influence Zone — area where it can spread influence.
- Calculate the Local Average Influence in that zone → IL(v).
- A node becomes a candidate if:
- $IP(v) > IL_0(v)$ → Meaning it's more influential than its surroundings.

PHASE II - INFLUENTIAL NODE SELECTION

- Build an Influence-BFS Tree from each user.
- This tree shows how far & fast a user's influence spreads.

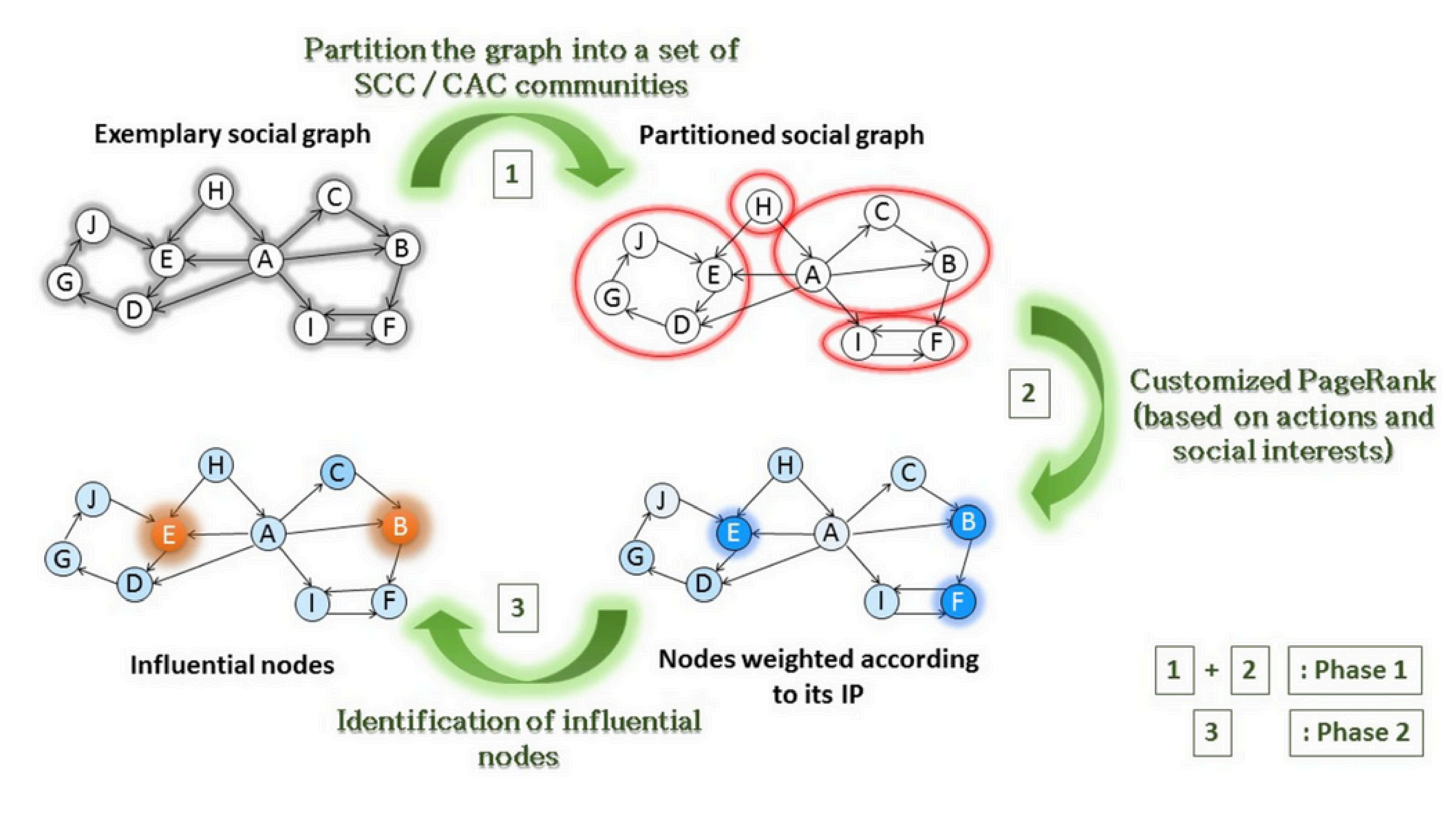
Choosing the Top Influencers

- Pick users with:
    - Highest influence score
    - Largest reach
    - Minimal overlap (so each spreads to a different audience)
- Stop when you have k best users.

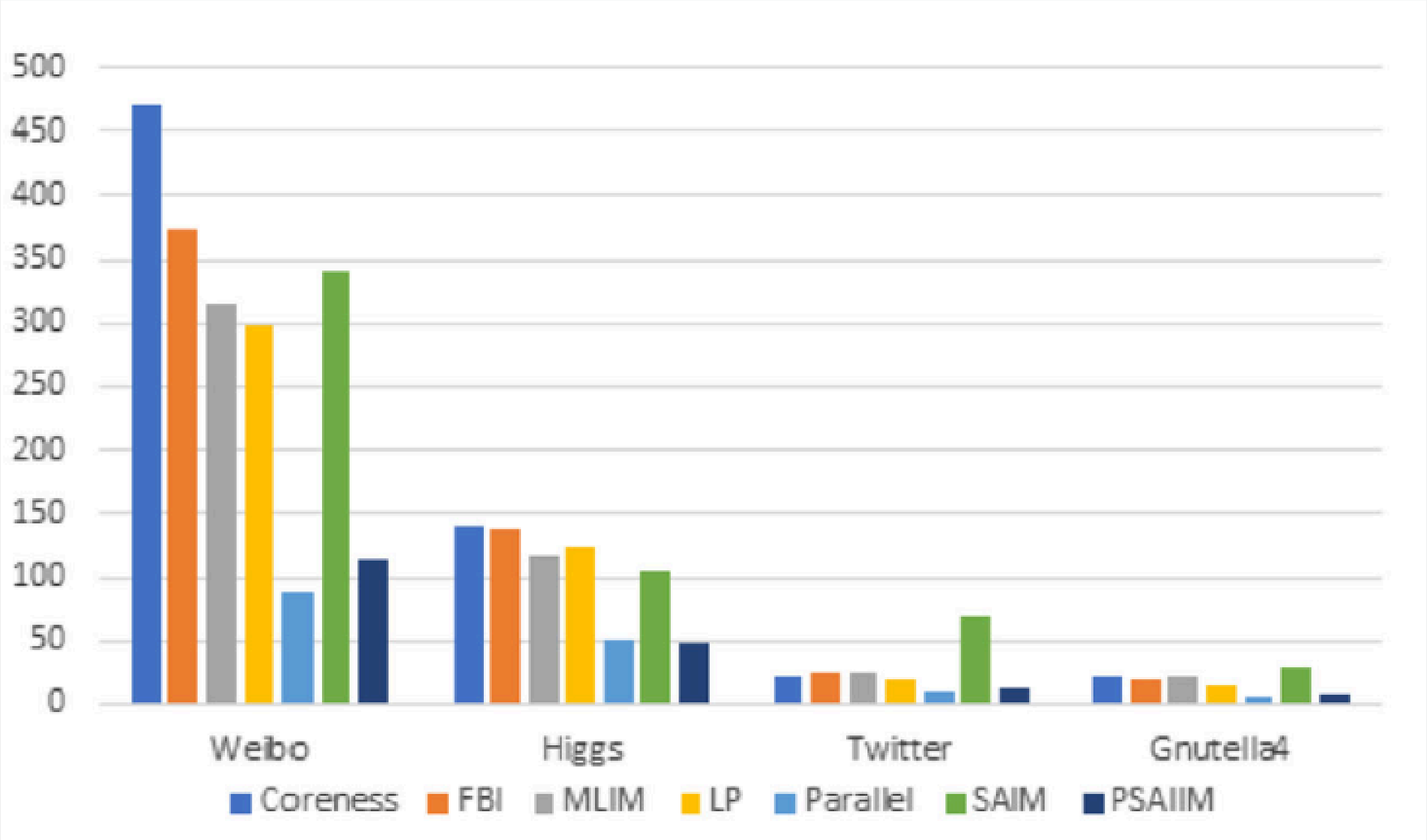
FINAL OUTPUT

- A list of top-k users who can spread information effectively.
- Fast and efficient thanks to:
  - Semantic data (interests, behavior)
  - Graph partitioning
  - Parallel processing

FLOW CHART OF THE PSAIIM ALGORITHM



RESULT



- Higher influence spread across small and mid-sized datasets due to its use of both social structure and semantic information
- On large datasets, its performance slightly declined as meaningful user interaction data diminished, but parallel speedup is more noticeable.

OUR IMPLEMENTATION

- We will be applying distributed computing by using virtual machines, they will communicate using MPI
- We will be using METIS for graph partitioning, and
- OpenMP for parallelization

USE OF METIS AND MPI

- The graph shall be subdivided into smaller graphs using METIS
- Each process on each virtual machine will apply PSAIIM algorithm on its subgraph
- The master node will send subgraphs to each of the processes
- After the processes on the machines have completed their processing, the results are gathered at master

USE OF OPENMP

- In each process, the influence power calculation phase takes place
- This is implemented using the PageRank algorithm, which is parallelized
- This parallelization is achieved using OpenMP-each thread is assigned a community

WHY OPENMP?

- OpenMP is optimized for CPU-based parallelism, which fits well for running on clusters with multi-core CPUs
- Since PSAIIM is graph-based and involves recursive structures like BFS trees, OpenMP allows you to parallelize loops easily without porting the entire algorithm to a GPU programming model (which OpenCL requires).

TEMPORAL COMPLEXITY ANALYSIS OF OUR IMPLEMENTATION

number of edges

number of nodes

$$O\left(\frac{m}{p} + n\right) + T_{comm}$$

number of threads

communication among processes

THANK YOU