

BUTTERFLY

BUTTERFLY



Understanding
Bipartite Graphs &
Motif Analysis

AHMED MURTAZA MALIK

HAMZA AHMAD MIR

AHMED HANNAN

COUNTING

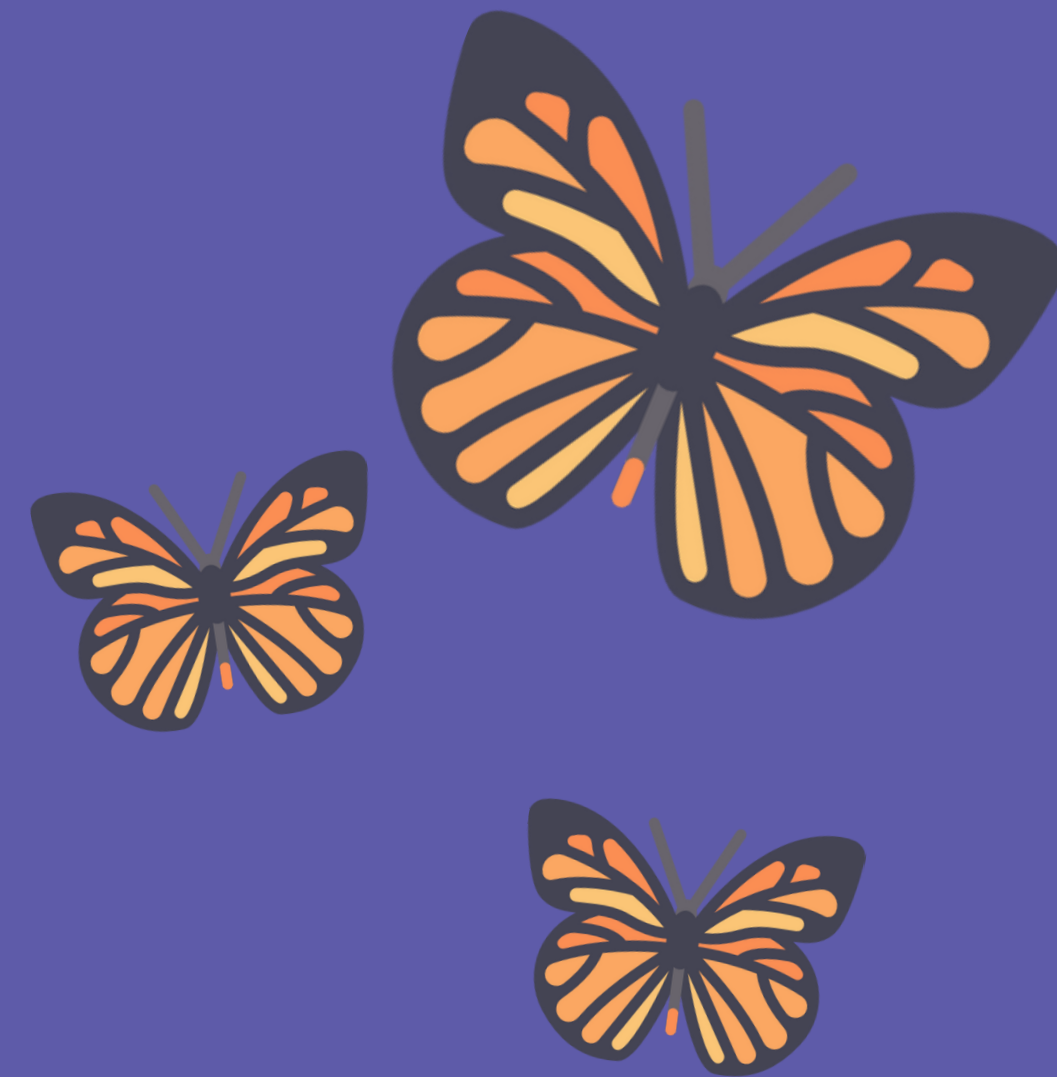
COUNTING

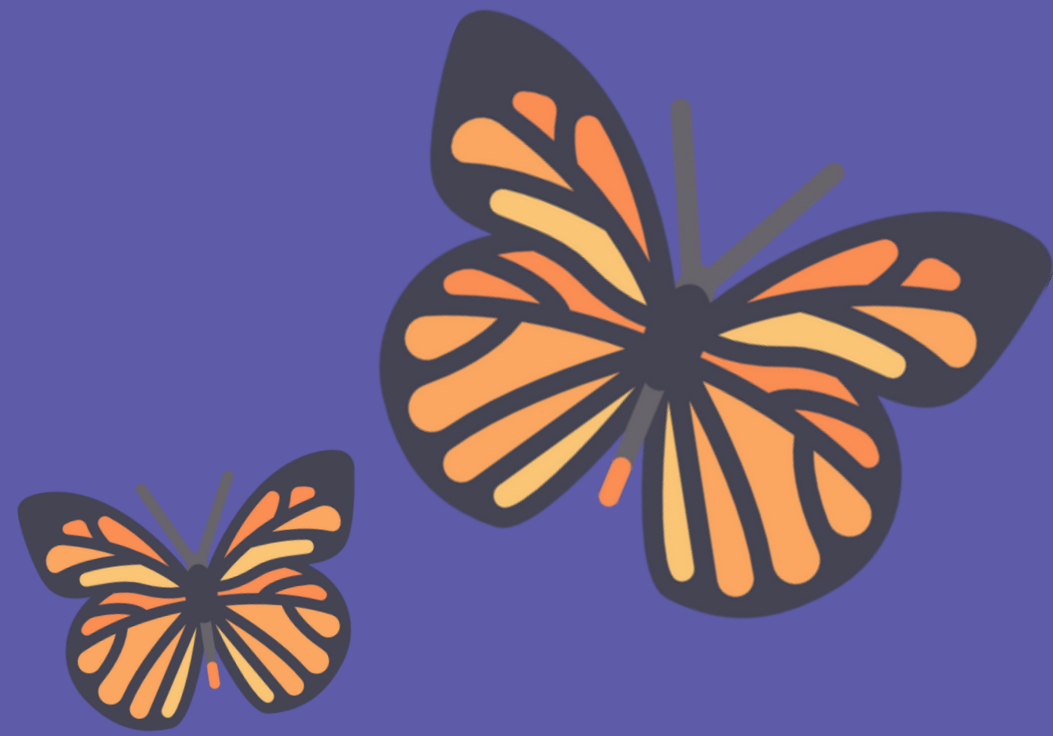
WHY BUTTERFLY COUNTING?

Helps in:

- Detecting fraud/spam
- Recommendation systems
- Understanding dense network patterns

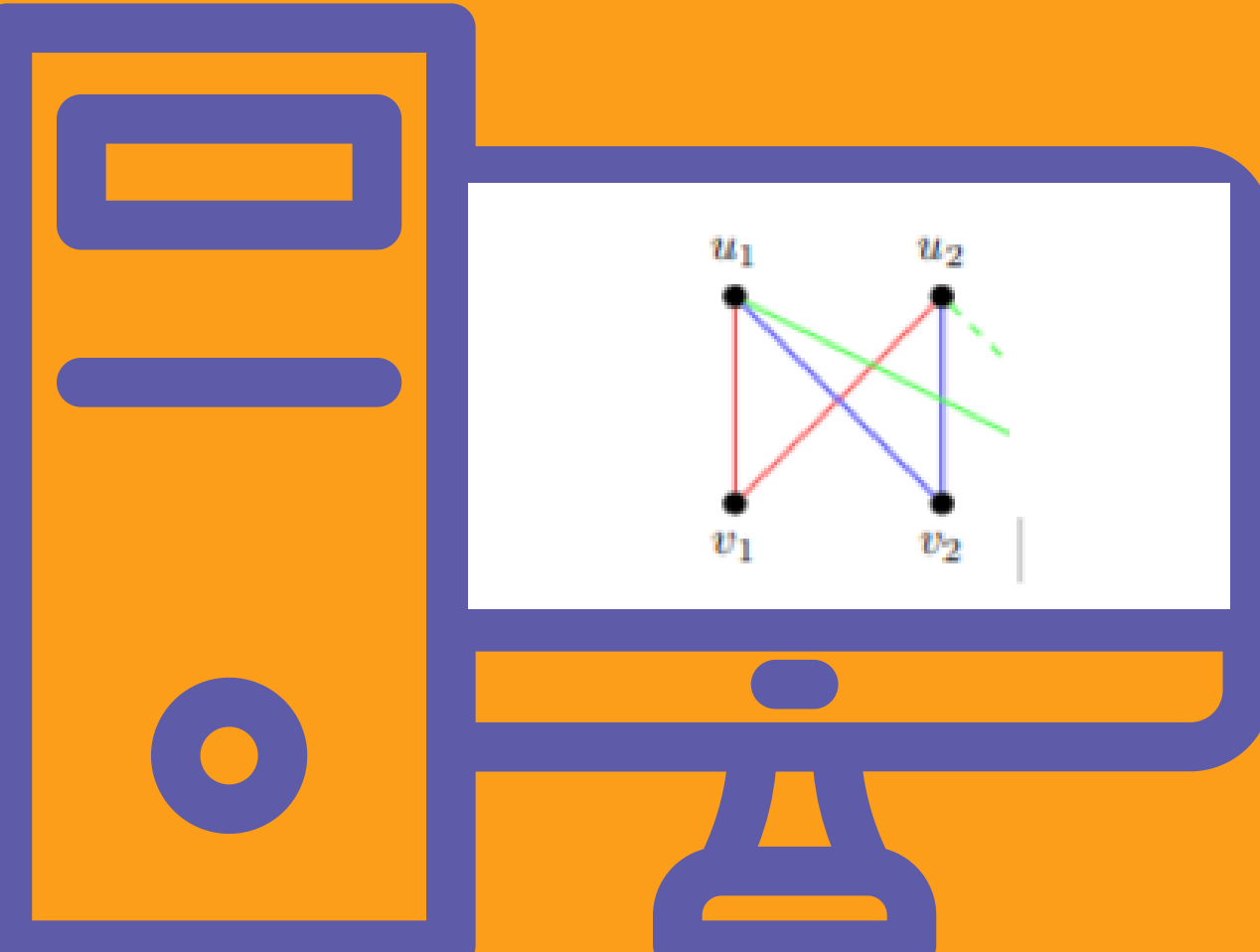
- Motif in Bipartite Graphs = Butterfly

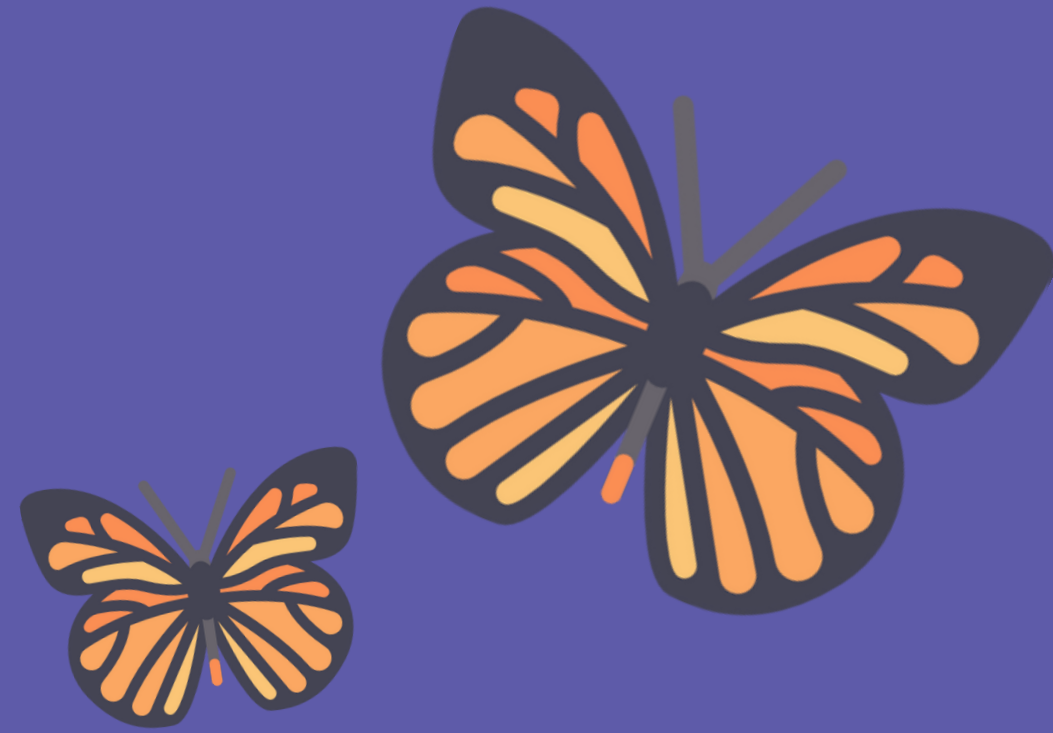




WHAT IS A BUTTERFLY?

- Bipartite Graph: Nodes in two sets (U & V)
- Butterfly:
 - 2 nodes in U connect fully to 2 nodes in V
- Wedge:
 - A path of 2 edges (used to count butterflies)

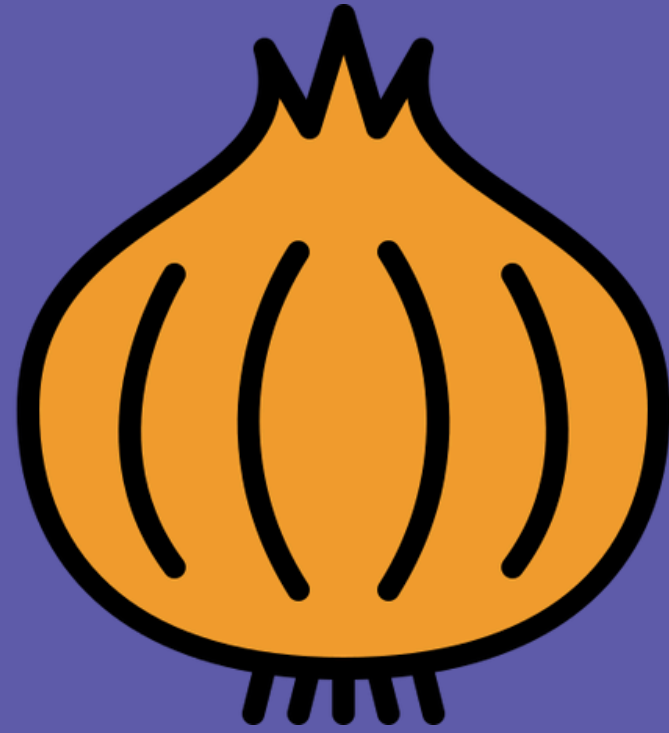




- Step 1: Rank vertices (by degree, coreness, etc.)
- Step 2: Retrieve wedges
- Step 3: Group wedges (Sorting, Hashing, Batching)
- Step 4: Count butterflies

PARBUTTERFLY WORKFLOW





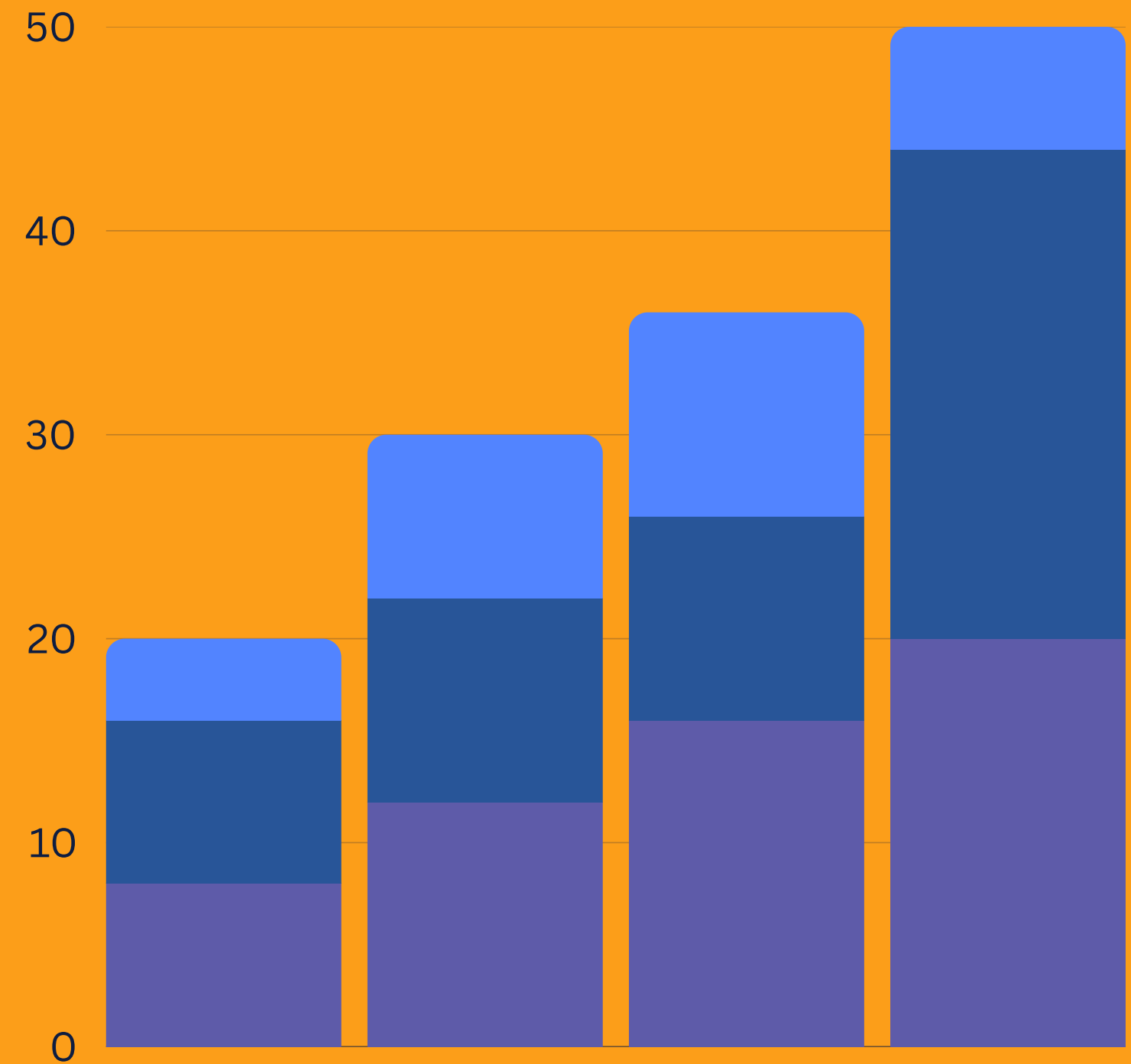
PEELING FOR CORE STRUCTURE

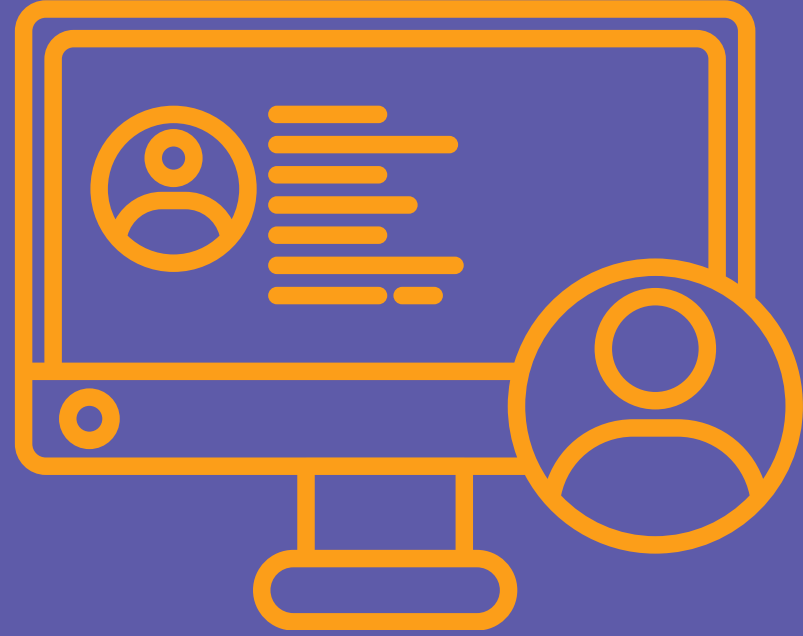
- Tip Peeling: Remove vertex with fewest butterflies
- Wing Peeling: Remove edge with fewest butterflies
- Output:
- Tip Number (vertex depth)
- Wing Number (edge depth)



ALGORITHMS & EFFICIENCY

Preprocessing
Counting Algorithms
Peeling Algorithms:
Work & Span





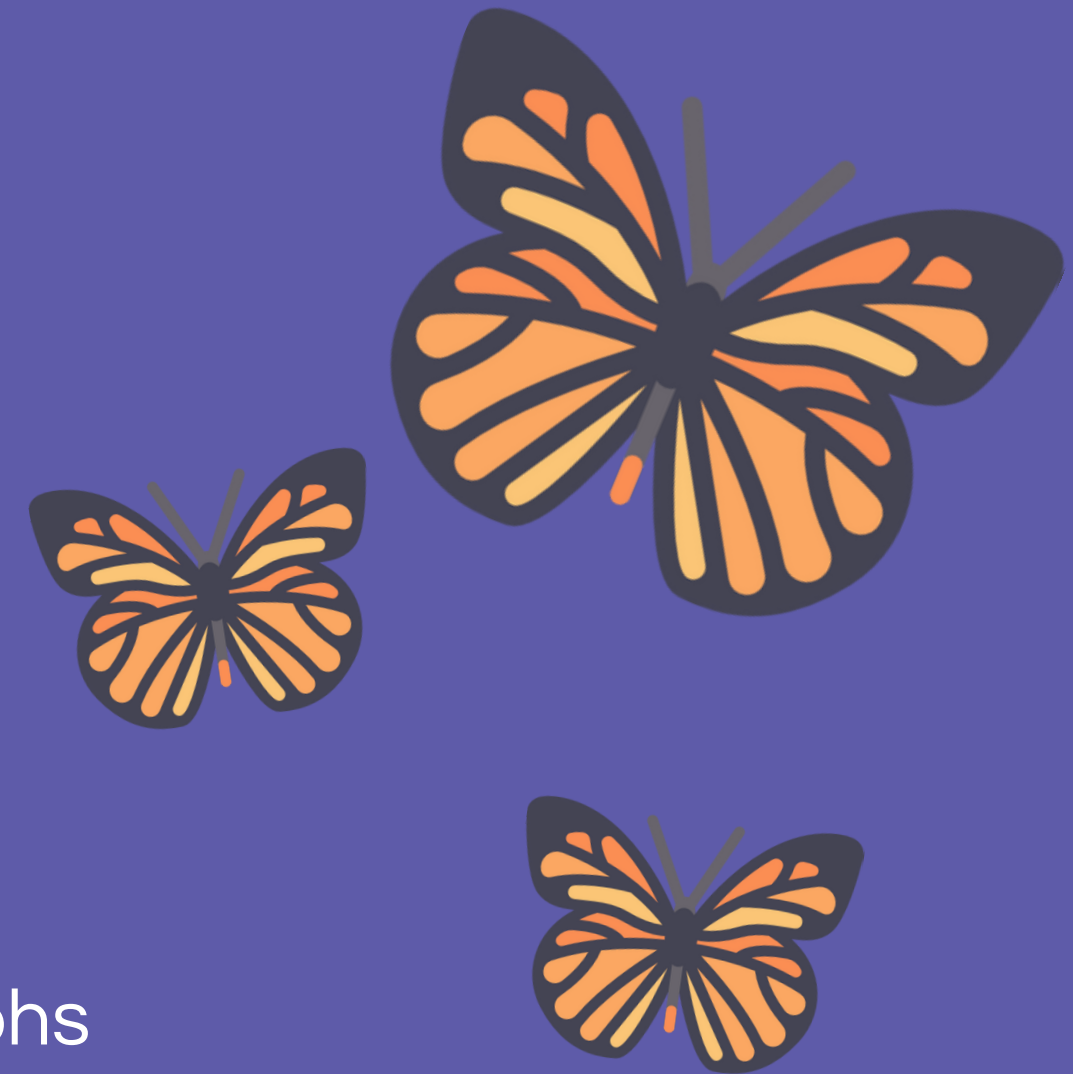
- System Used: AWS 48-core, 384 GB RAM
- Speedups:
 - 13.6× over sequential
 - 5169× over older parallel tools
- Best techniques:
 - Batching for aggregation
 - Side-order ranking

RESULTS & SPEEDUPS

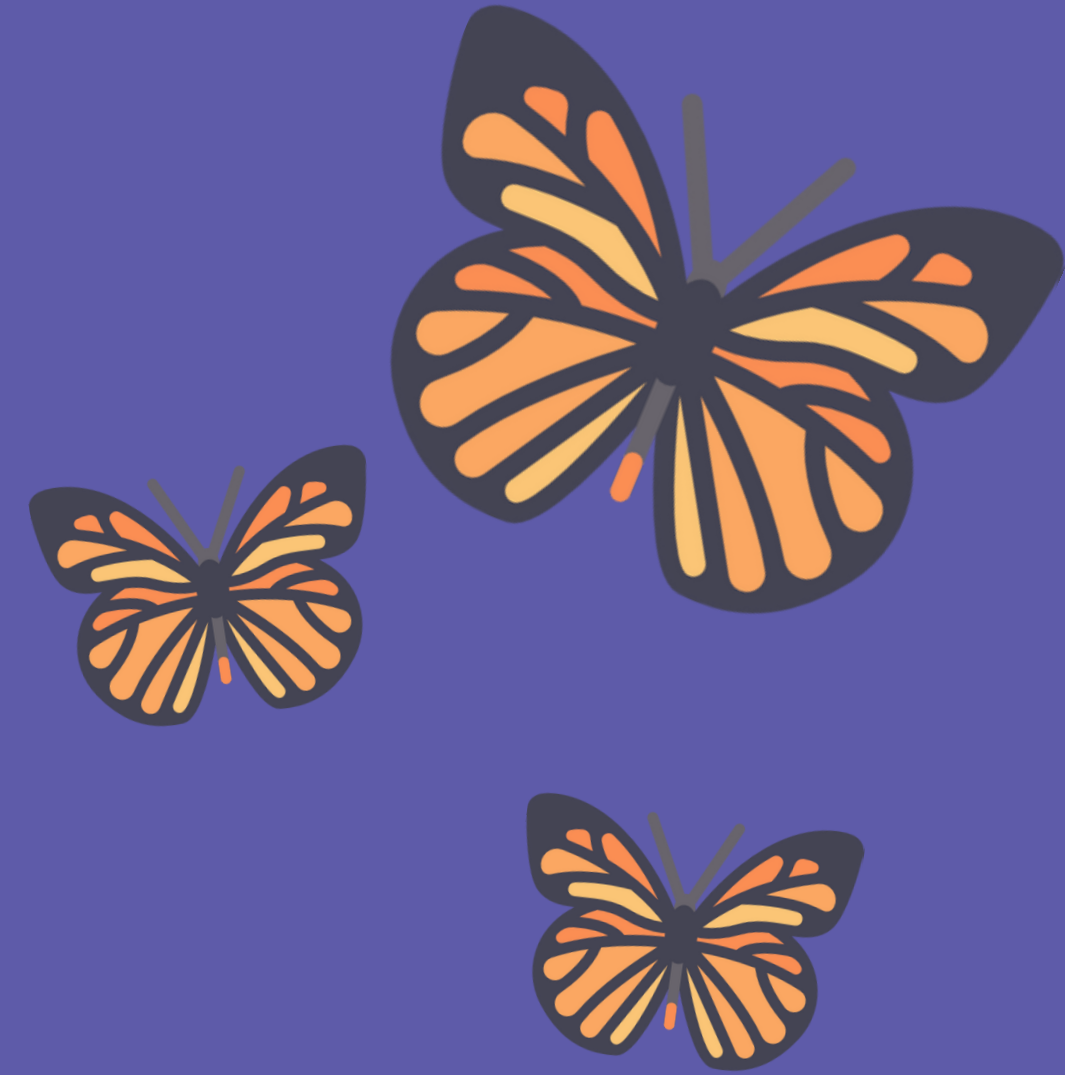
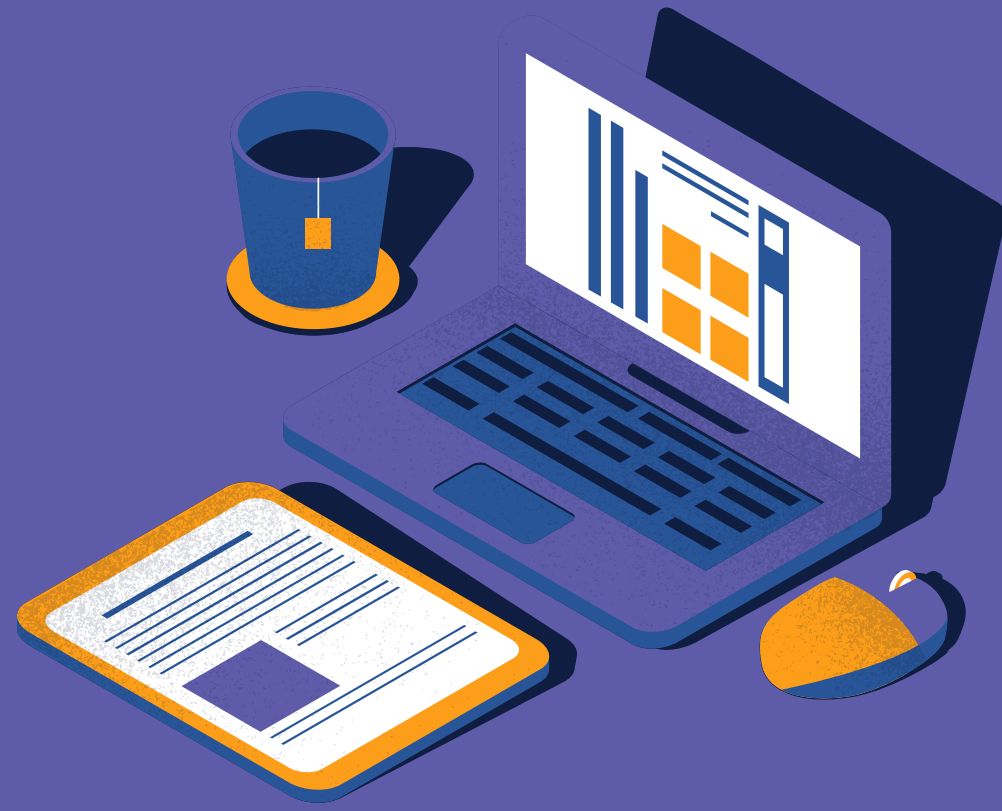


CONCLUSION

- ParButterfly is:
- Fast
- Scalable
- Accurate (even approx. counting)
- Enables efficient motif analysis on huge bipartite graphs



GitHub Repo Link: <https://github.com/ahmedmurtazamalik/butterfly-parallelization>



THANK YOU!

22I-0943

22I-0985

22I-1003