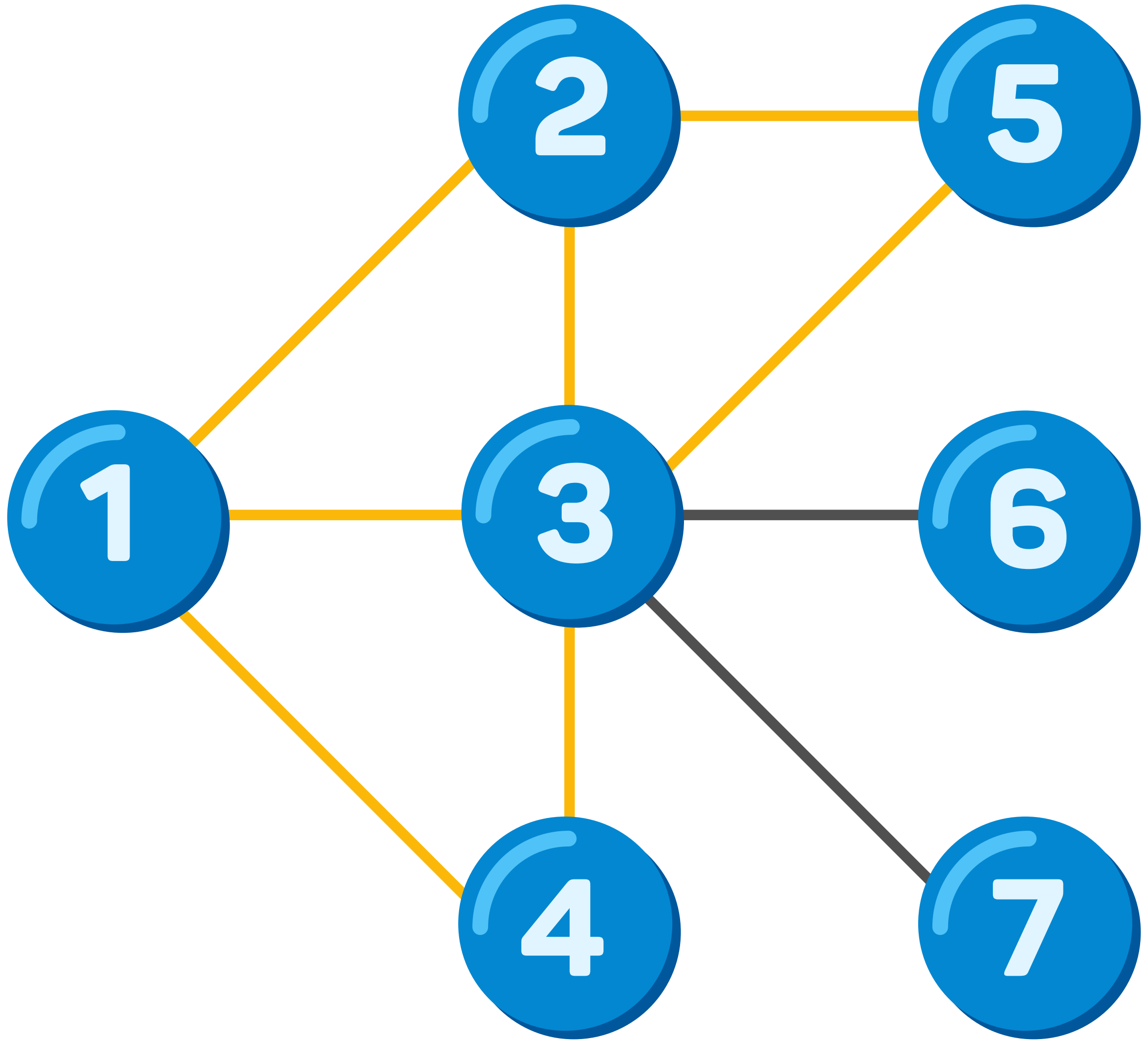
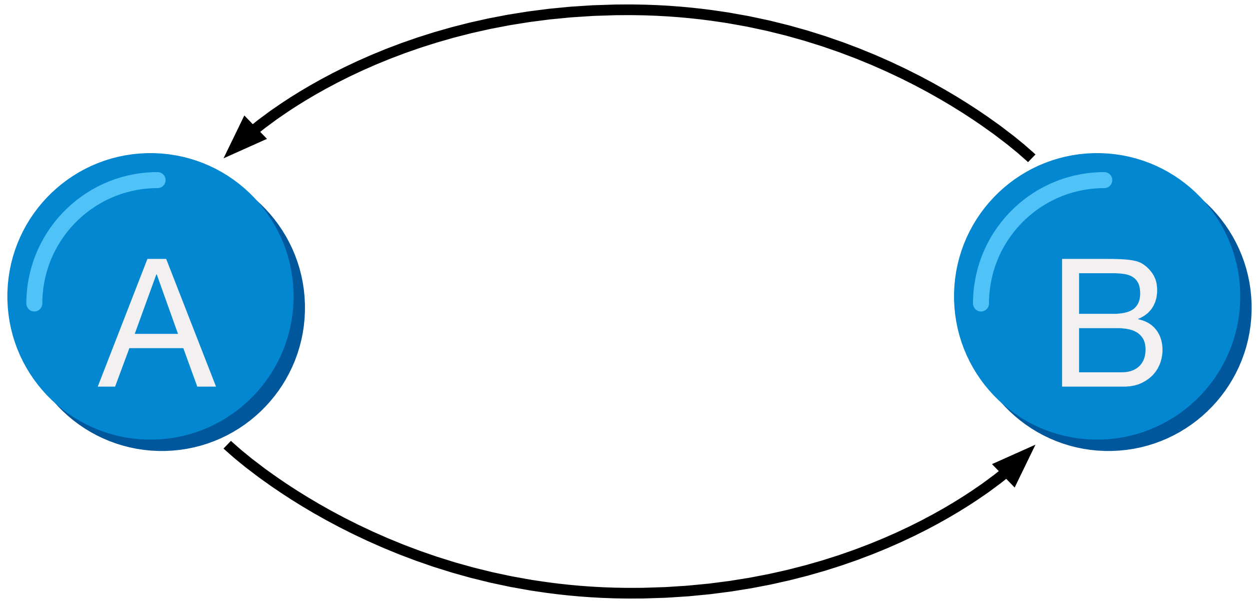
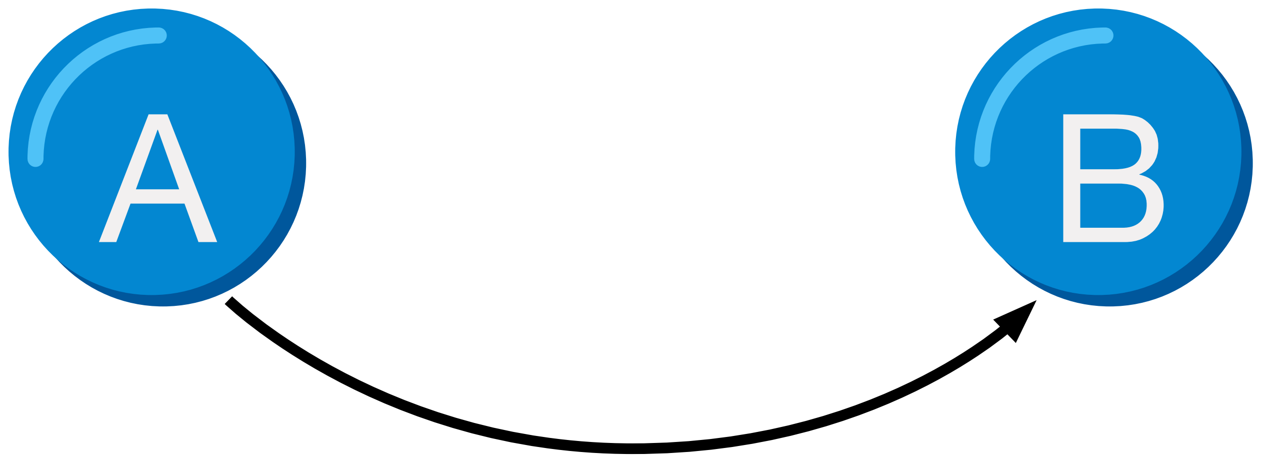
GraphFrame trianglesCount: under the hood

Mini social graph

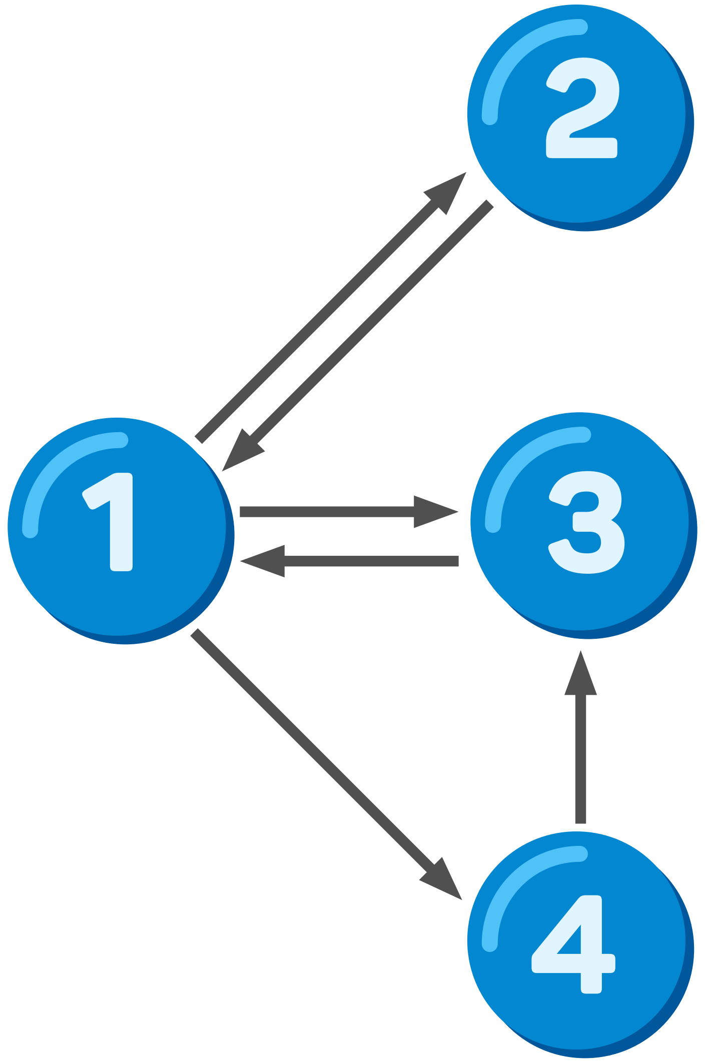


1 - 2 triangles
2 - 2 triangles
3 - 3 triangles
4 - 1 triangle
5 - 1 triangle
6 - 0 triangles
7 - 0 triangles


A
B


A
B

Flipping edges



(1, 2) → (1, 2)

(2, 1)

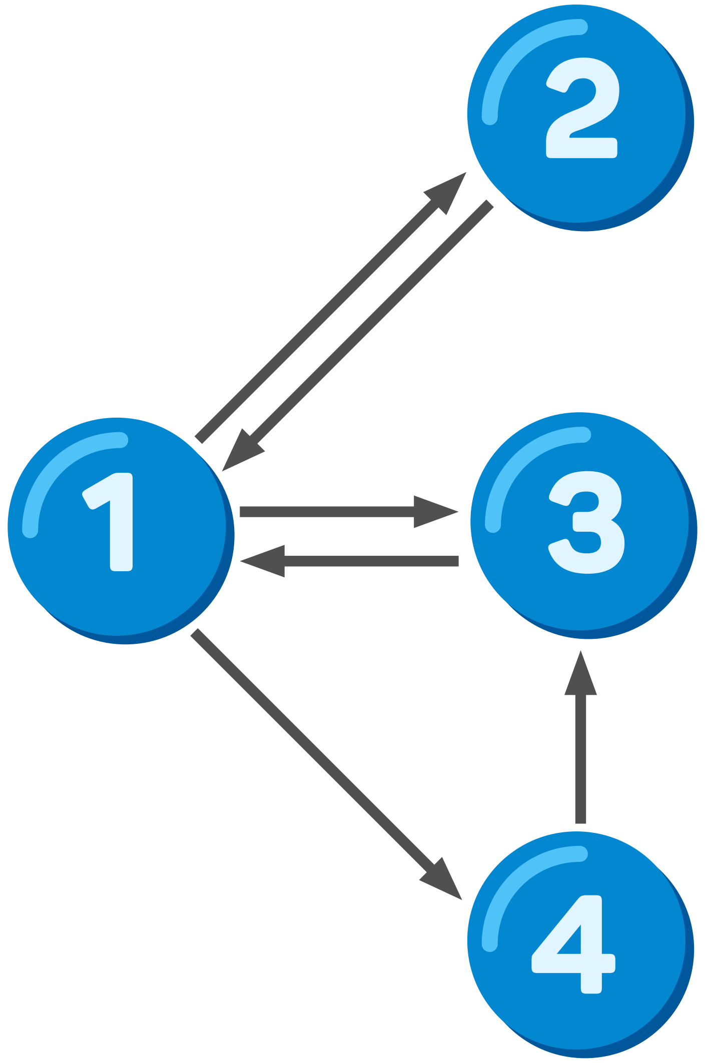
(1, 3)

(3, 1)

(1, 4)

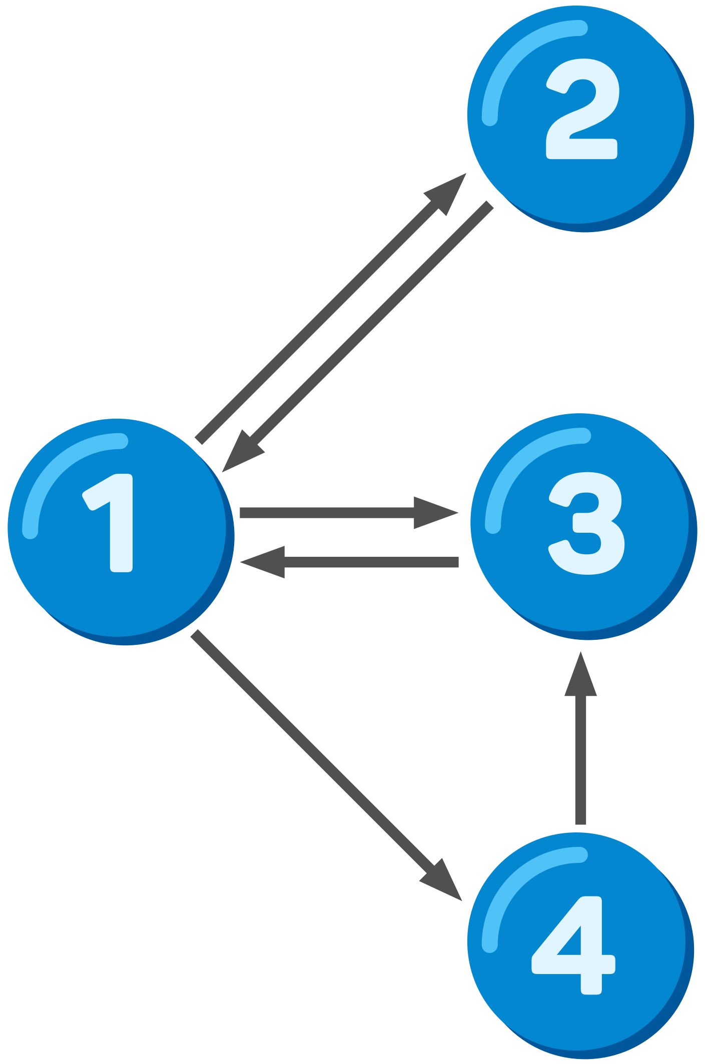
(4, 3)

Flipping edges



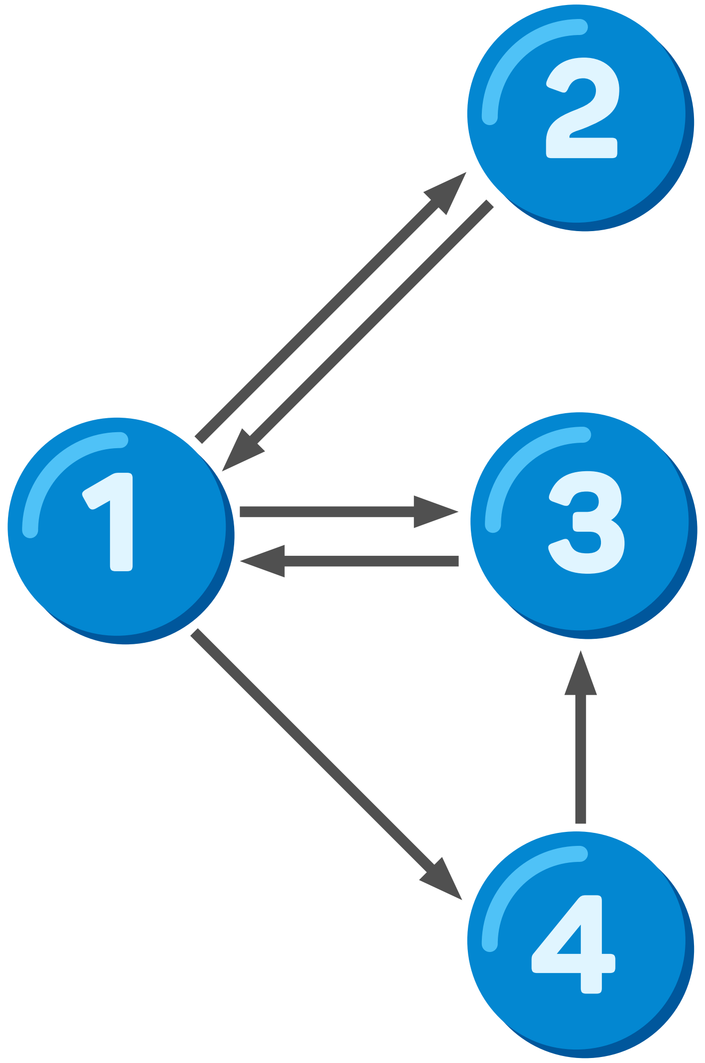
(1, 2)		(1, 2)
(2, 1)	→	(1, 2)
(1, 3)		
(3, 1)		
(1, 4)		
(4, 3)		

Flipping edges



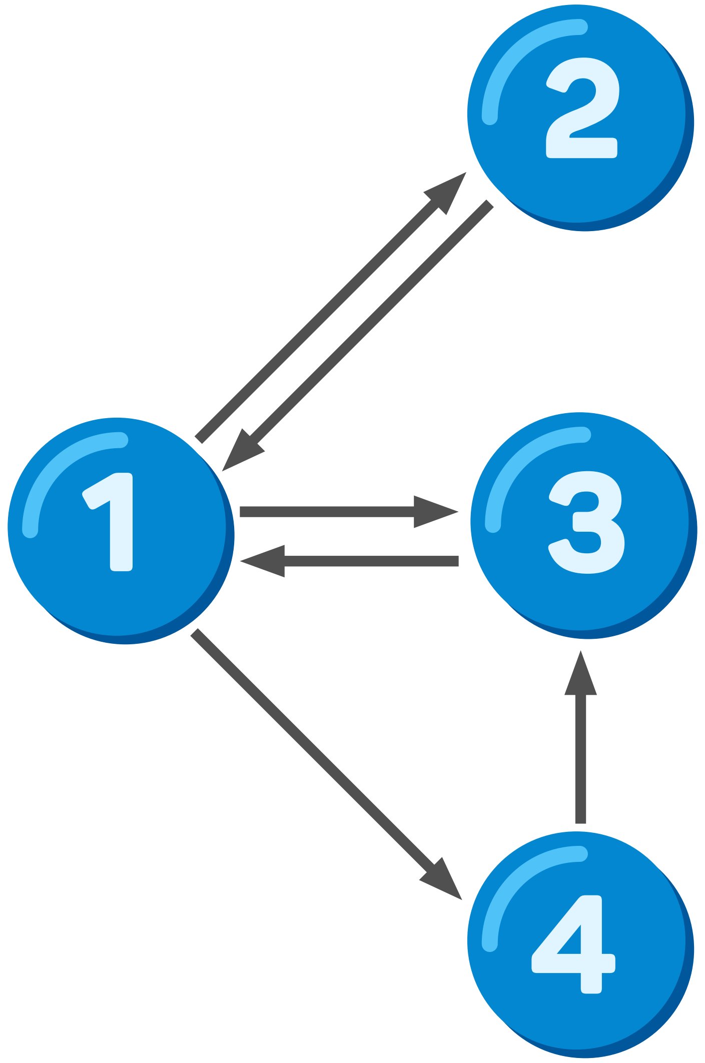
(1, 2)		(1, 2)
(2, 1)		(1, 2)
(1, 3)	→	(1, 3)
(3, 1)		
(1, 4)		
(4, 3)		

Flipping edges



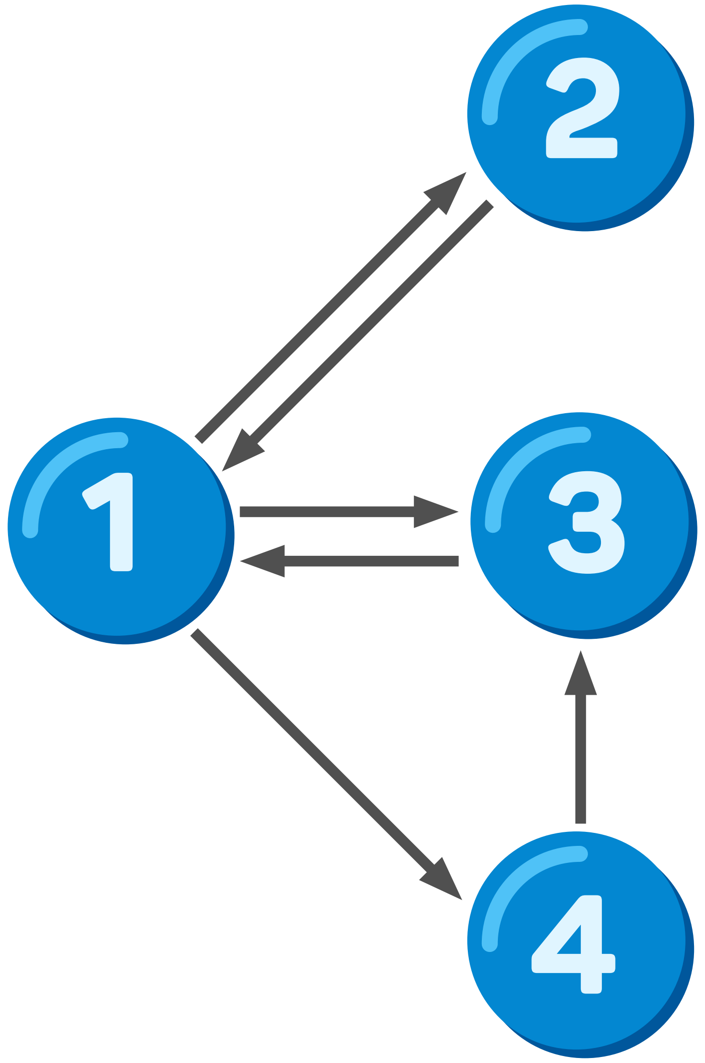
(1, 2)		(1, 2)
(2, 1)		(1, 2)
(1, 3)		(1, 3)
(3, 1)	→	(1, 3)
(1, 4)		
(4, 3)		

Flipping edges



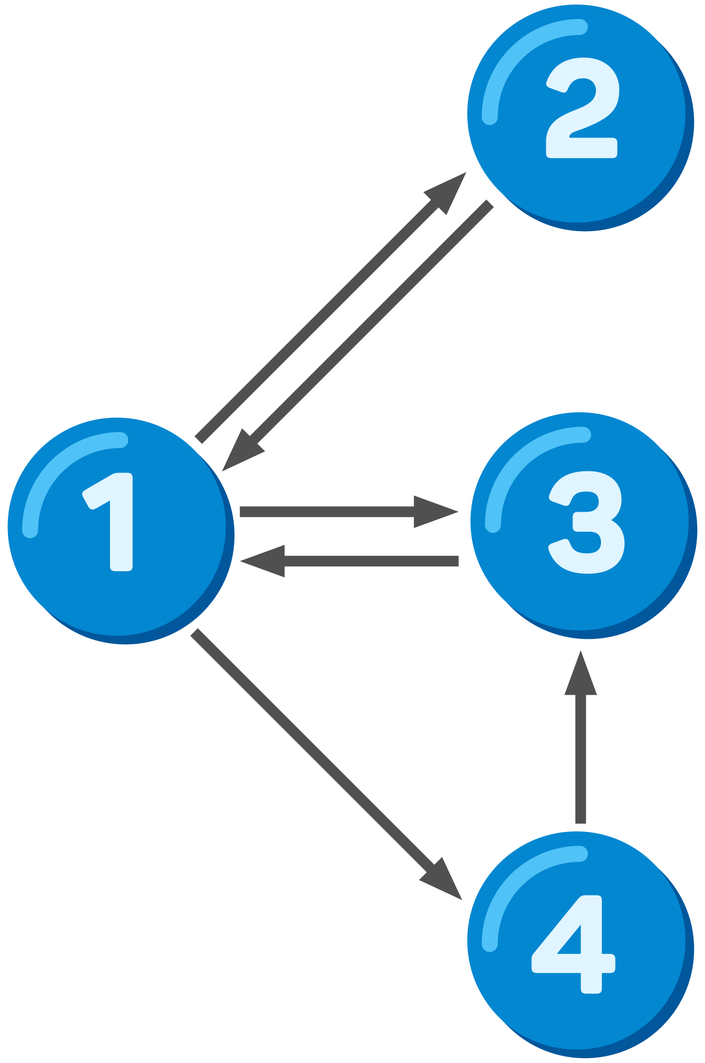
(1, 2)		(1, 2)
(2, 1)		(1, 2)
(1, 3)		(1, 3)
(3, 1)		(1, 3)
(1, 4)	→	(1, 4)
(4, 3)		

Flipping edges



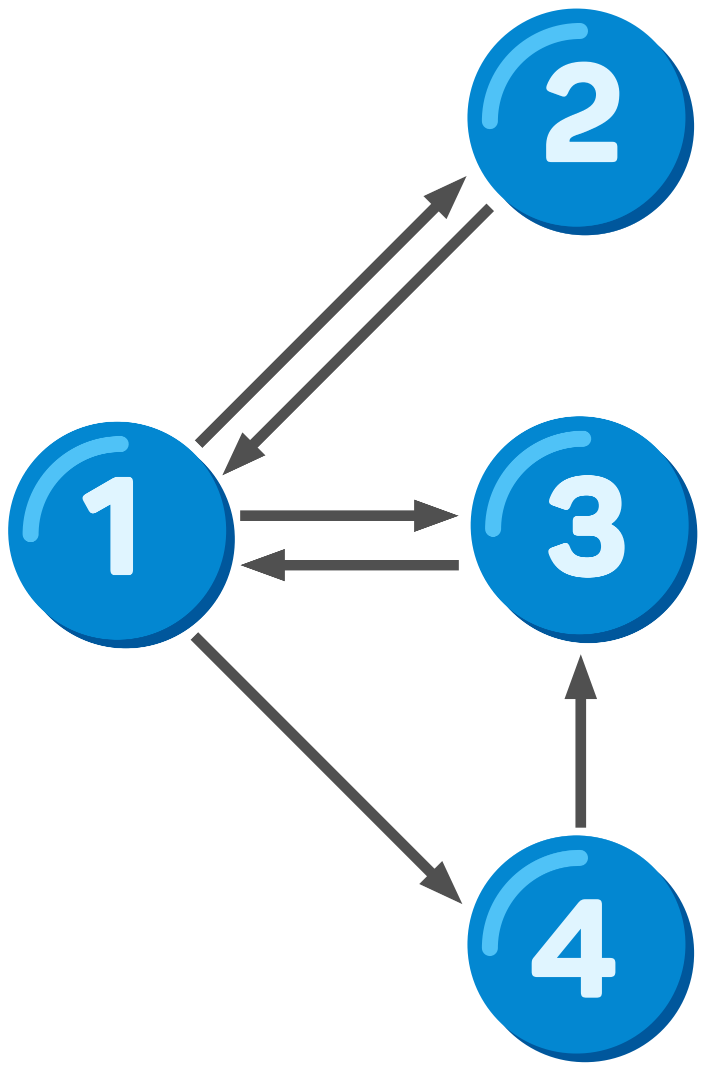
(1, 2)		(1, 2)
(2, 1)		(1, 2)
(1, 3)		(1, 3)
(3, 1)		(1, 3)
(1, 4)		(1, 4)
(4, 3)	→	(3, 4)

Flipping edges



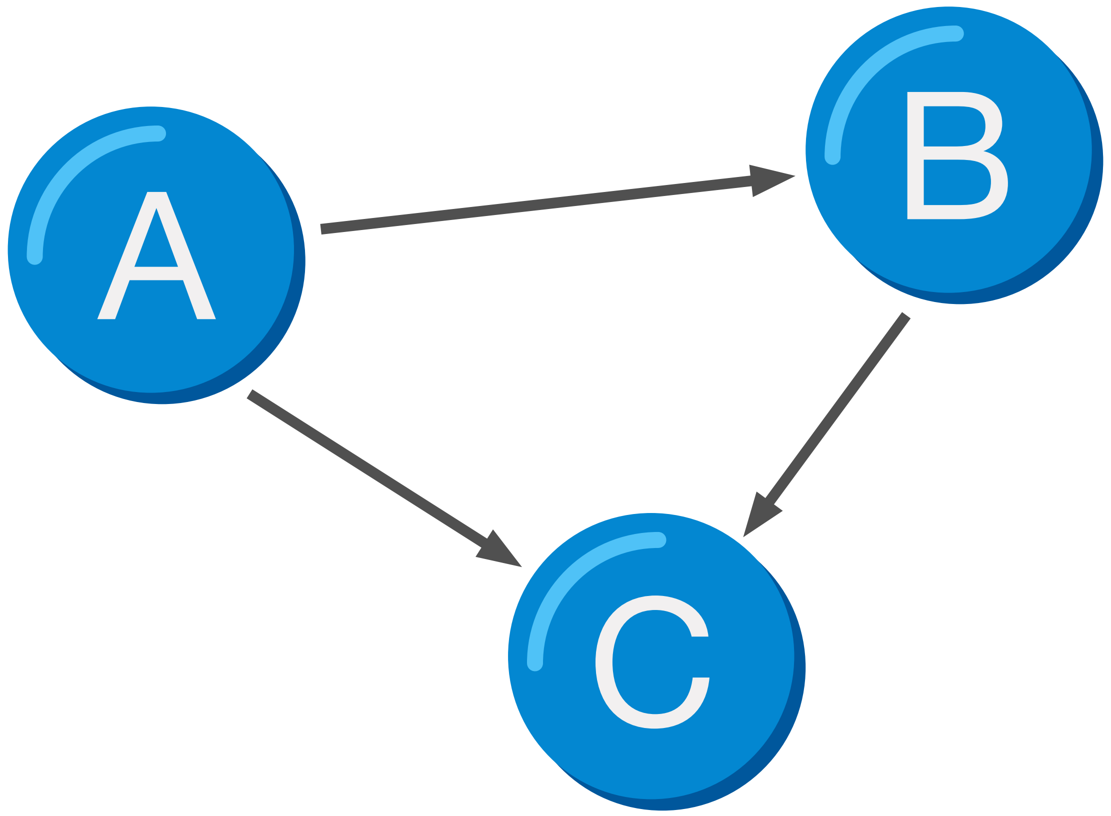
(1, 2)		(1, 2)
(2, 1)		(1, 2)
(1, 3)	↦	(1, 3)
(3, 1)		(1, 3)
(1, 4)		(1, 4)
(4, 3)		(3, 4)

Flipping edges

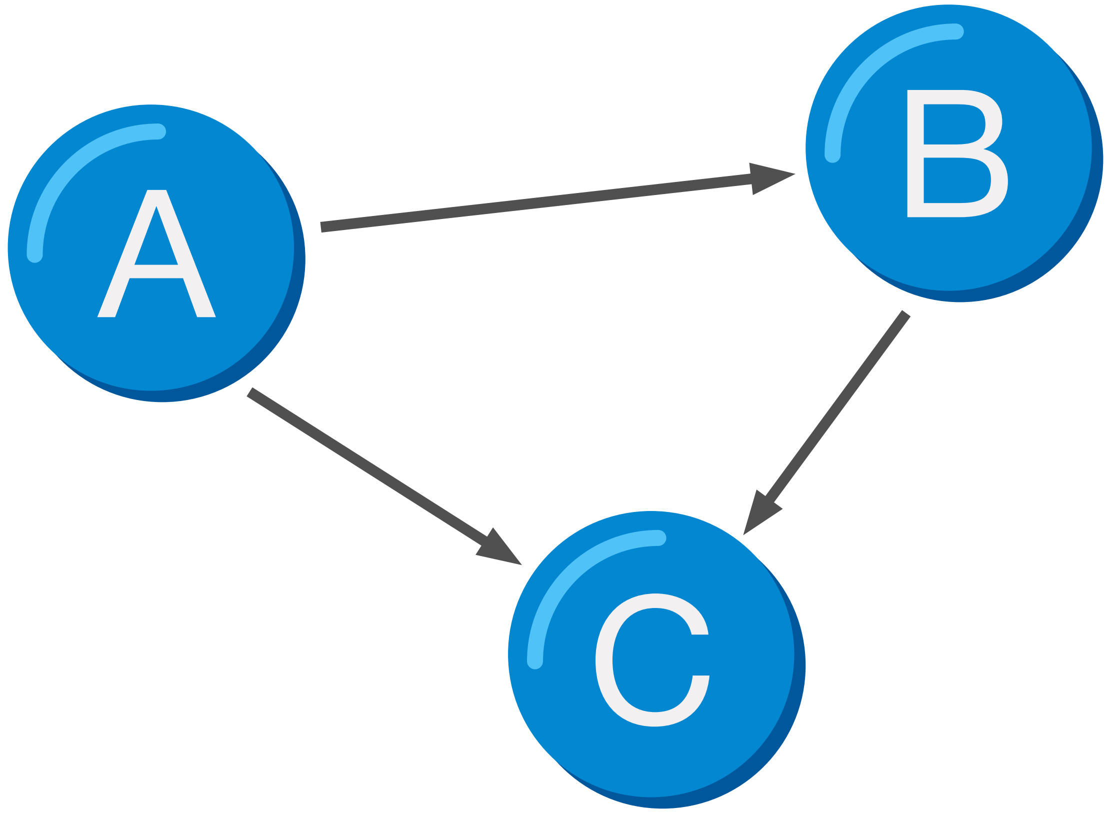


(1, 2)		(1, 2)		(1, 2)
(2, 1)		(1, 2)		(1, 3)
(1, 3)	↦	(1, 3)	↦	(1, 4)
(3, 1)		(1, 3)		(3, 4)
(1, 4)		(1, 4)		
(4, 3)		(3, 4)		

Triangle - a set of 3 vertices, provided there is an edge between any 2 of them.

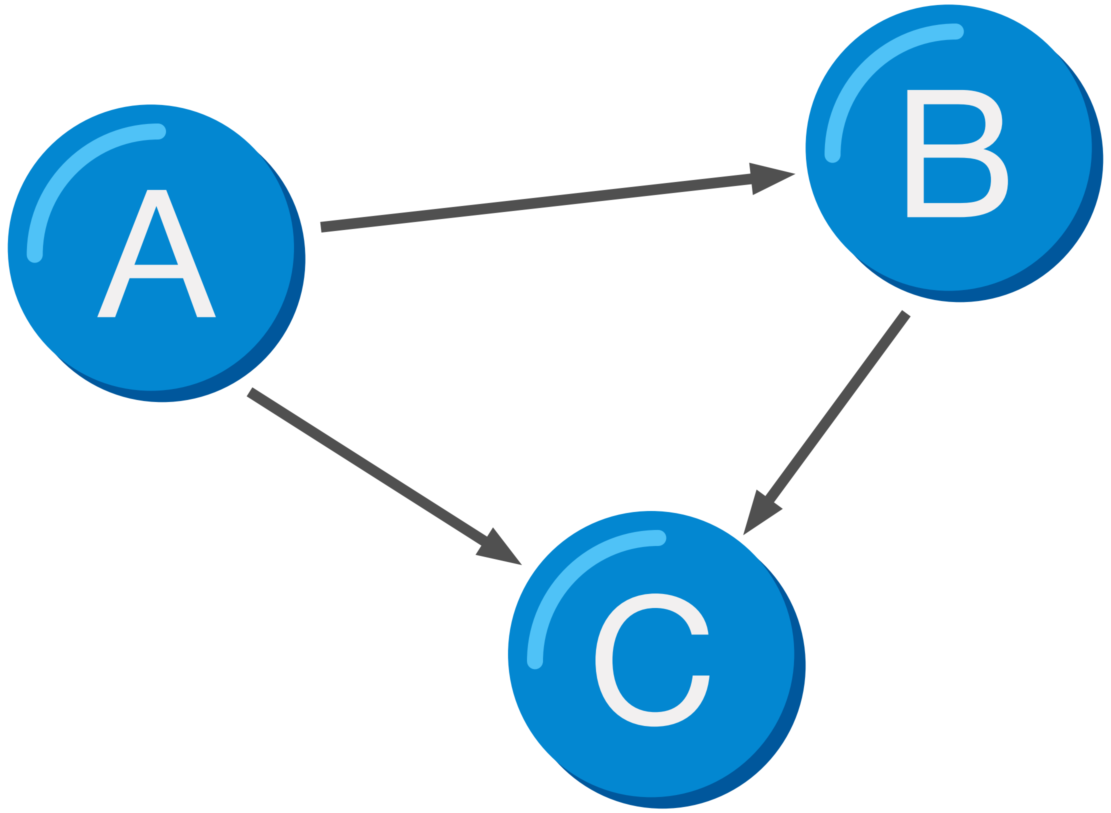


Triangle - a set of 3 vertices, provided there is an edge between any 2 of them.



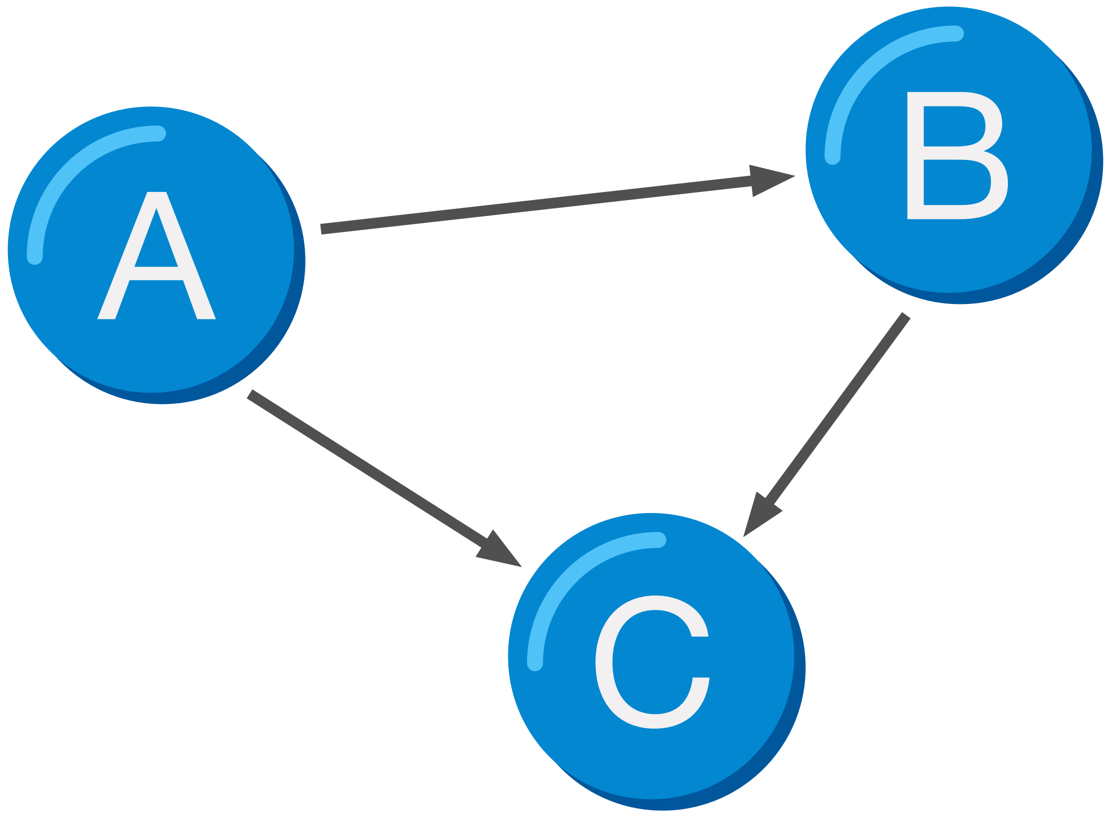
(A, B): A < B
(B, C): B < C

Triangle - a set of 3 vertices, provided there is an edge between any 2 of them.



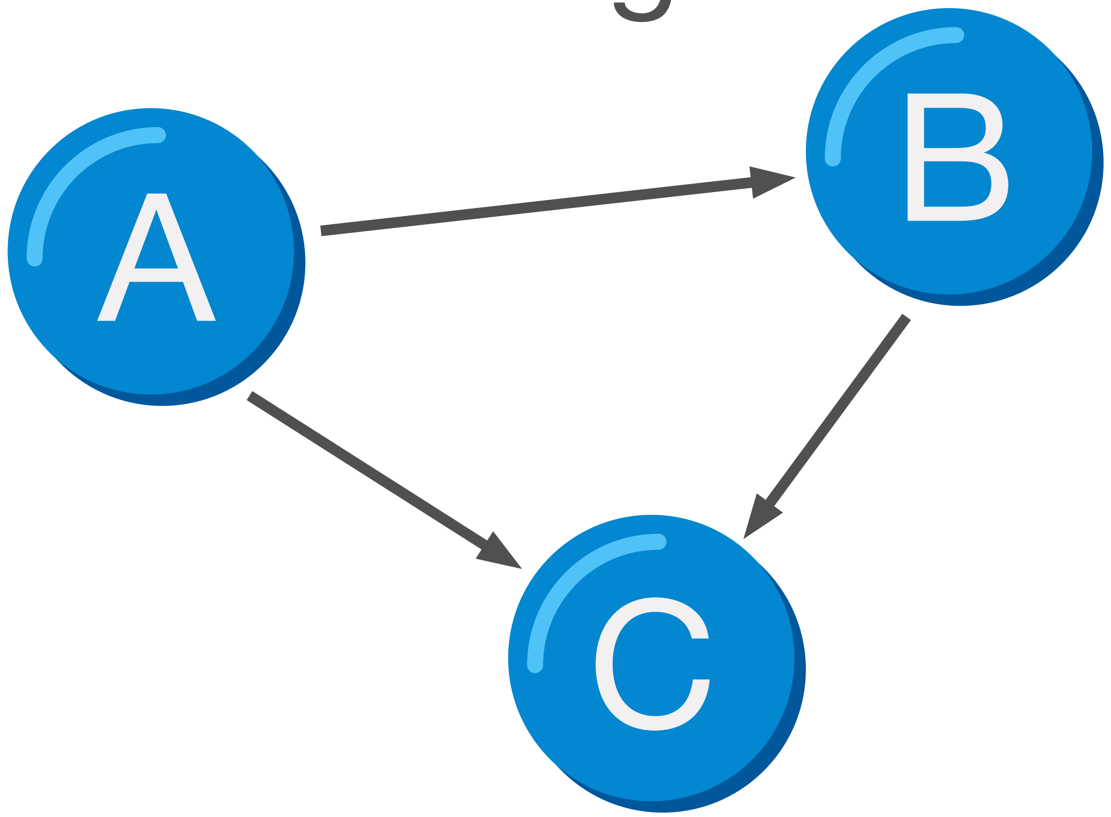
$$\begin{matrix}(A, B): A < B \\ (B, C): B < C\end{matrix} \longrightarrow A < C$$

Triangle - a set of 3 vertices, provided there is an edge between any 2 of them.



(A, B): A < B
(B, C): B < C → A < C → (A, C)

Triangle



Motif finding DSL

(A)-[]->(B); (B)-[]->(C); (A)-[]->(C)

Motif finding DSL

(A)-[]->(B); (B)-[]->(C); (A)-[]->(C)

```
triangles = g.find("(A)-[]->(B); (B)-[]->(C); (A)-[]->(C)")
triangles.show()
```

```
+--------------------+--------------------+--------------------+
|                   A|                   B|                   C|
+--------------------+--------------------+--------------------+
|  [1,Alex,28,M,MIPT]| [2,Emeli,28,F,MIPT]|[3,Natasha,27,F,S...|
|  [1,Alex,28,M,MIPT]|[3,Natasha,27,F,S...| [4,Pavel,30,M,MIPT]|
| [2,Emeli,28,F,MIPT]|[3,Natasha,27,F,S...|  [5,Oleg,35,M,MIPT]|
+--------------------+--------------------+--------------------+
```

Explode triangles DF

```
vertexTriangles = triangles.selectExpr("A.id as A", "B.id as B", "C.id as C") \
    .select(array(col("A"), col("B"), col("C")).alias("triangleVertices")) \
    .select(explode("triangleVertices").alias("id")) \
    .groupBy("id") \
    .count()

vertexTriangles.show()
```

```
+---+-----+
| id|count|
+---+-----+
|  1|    2|
|  2|    2|
|  3|    3|
|  4|    1|
|  5|    1|
+---+-----+
```

Join vetrexTriangles with original info

```
g.vertices.join(vertexTriangles, "id", "left_outer").show()
```

```
+---+-------+---+------+----------+-----+
| id|   name|age|gender|university|count|
+---+-------+---+------+----------+-----+
|  1|   Alex| 28|     M|      MIPT|    2|
|  2|  Emeli| 28|     F|      MIPT|    2|
|  3|Natasha| 27|     F|     SPbSU|    3|
|  4|  Pavel| 30|     M|      MIPT|    1|
|  5|   Oleg| 35|     M|      MIPT|    1|
|  6|   Ivan| 30|     M|       MSU| null|
|  7|   Ilya| 29|     M|       MSU| null|
+---+-------+---+------+----------+-----+
```

TrianglesCount algorithm

1. Flip all edges in such way as src < dst and delete all duplicates

2. Find all triangles with motif using pattern “(A)-[]->(B); (B)-[]->(C); (A)-[]->(C)”

3. Explode triangles and count occurrence of each vertex

4. Join info about triangles for each vertex with original info about it

TrianglesCount algorithm

1. Flip all edges in such way as src < dst and delete all duplicates
   0 shuffles
2. Find all triangles with motif using pattern “(A)-[]->(B); (B)-[]->(C); (A)-[]->(C)”
   6 shuffles
3. Explode triangles and count occurrence of each vertex
   1 shuffle
4. Join info about triangles for each vertex with original info about it
   1 shuffle

8 shuffles

Summary

- You have learned how triangleCount method of GraphFrames works step by step

Summary

- You have learned how triangleCount method of GraphFrames works step by step
- You have known how to estimate complexity of Graph Frames triangle count algorithm implementation in terms of shuffles