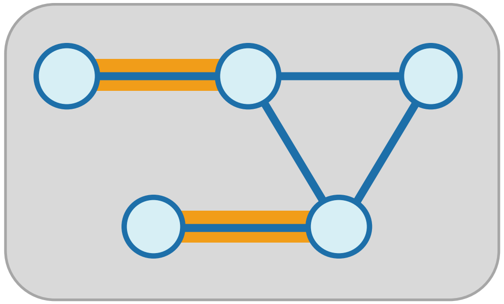
A $(2+\epsilon)$-Approximation for Maximum Weight Matching in the Semi-Streaming Model

Ami Paz (IRIF, Paris)
Joint work with Gregory Schwartzman (NII, Tokyo)

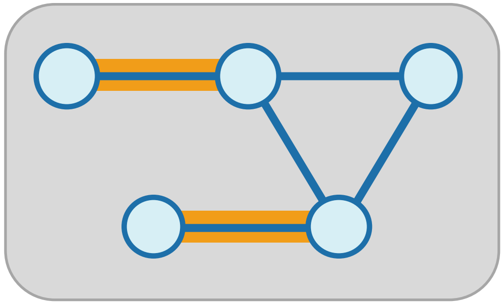
Matching

A set of non-adjacent edges



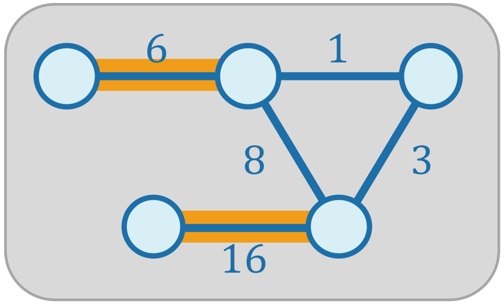
Maximal Matching

A matching that cannot be extended



Maximum Weight Matching

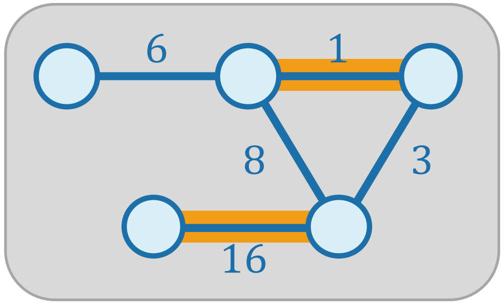
A matching of maximum weight



Total: **22**

Maximum Weight Matching

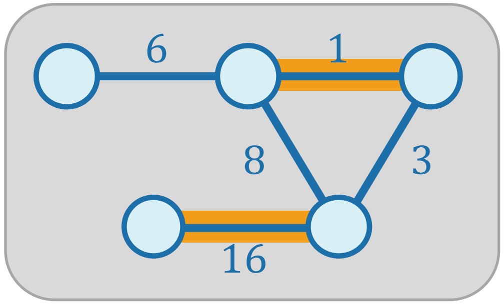
A matching of maximum weight



Total: 17

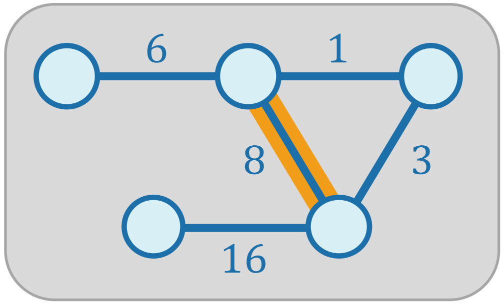
The Semi-Streaming Model

- Weighted graph, n vertices
- Stream of edges
  - Arbitrary order
  - Output after stream end
- Memory: $O(n \cdot \mathrm{polylog}\, n)$
- Single pass, insertion only
- Goal: good approximation ratio



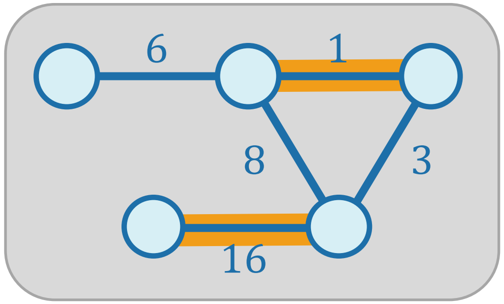
Total: 17

Example: Greedy



Output

Example: Greedy



Output

Known Results: Weighted

Approx. ratio	Authors
6	[FKMSZ'05]
5.828	[McGregor'05]
5.585	[Zelke'12]
$(4.911 + \epsilon)$	[ELMS'11]
$(4 + \epsilon)$	[CS'14]
$(\mathbf{2} + \boldsymbol{\epsilon})$	Current work

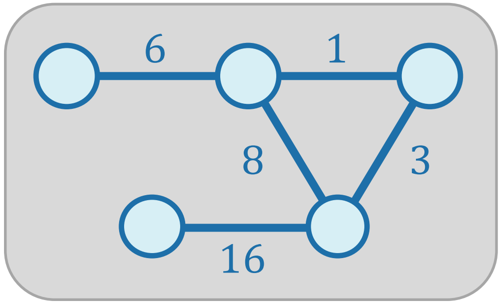
Deterministic, single pass

Known Results: Unweighted

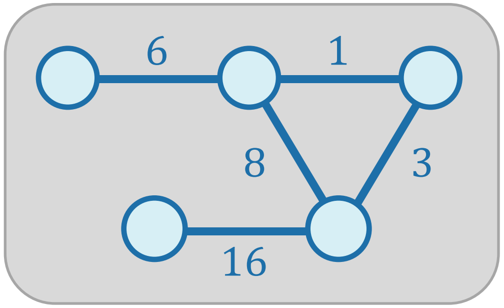
Approx. ratio	
2	Greedy [FKMSZ'05]
LB: 1.58	[Kapralov'12]

Better approximation for weighted

$\Longrightarrow$ Better approximation for unweighted

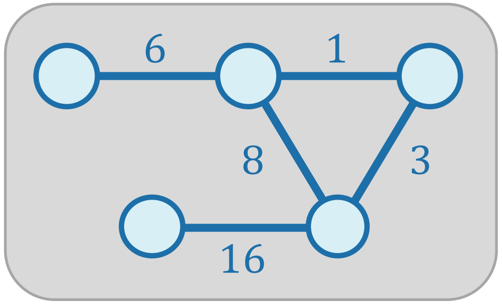


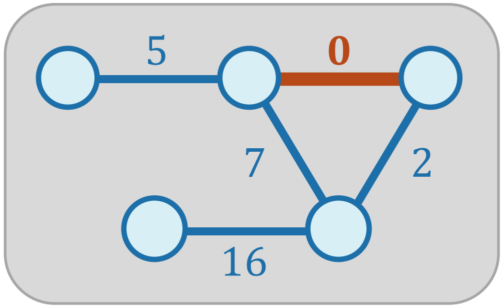
Sequential Local-Ratio



[Bar-Yehuda & Even '85]

Sequential Local-Ratio



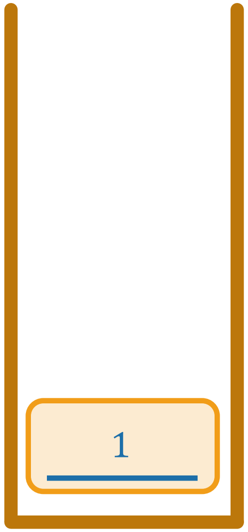


Pick an arbitrary edge e

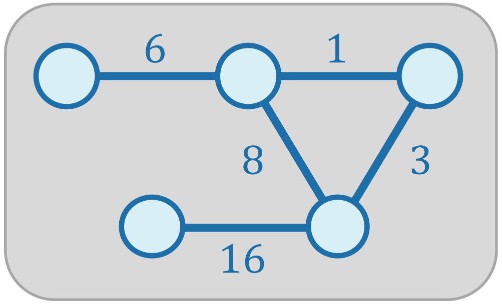
Push onto a stack

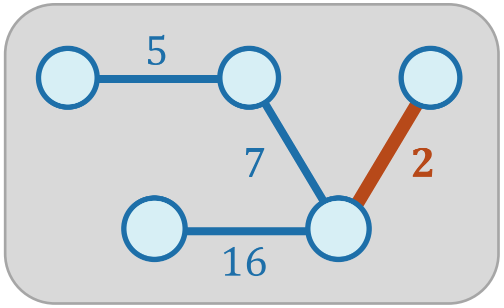
Reduce $w(e)$ from e's neighborhood

Remove non-positive edges

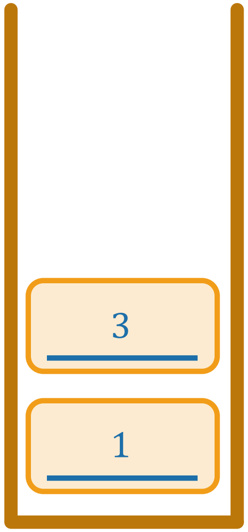


Sequential Local-Ratio

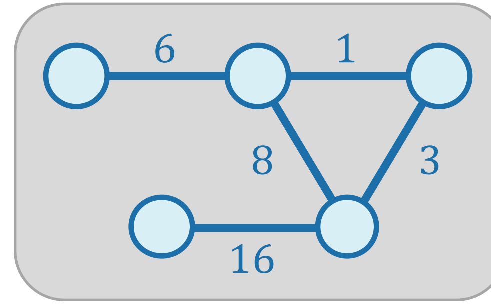


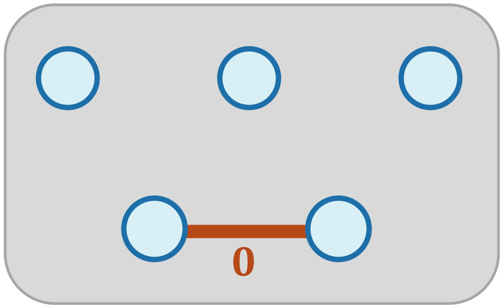


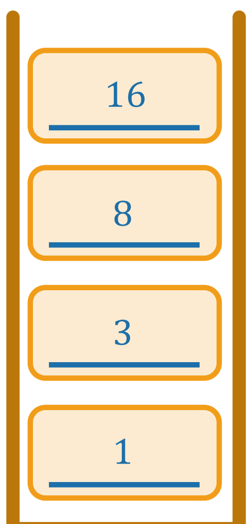
Repeat



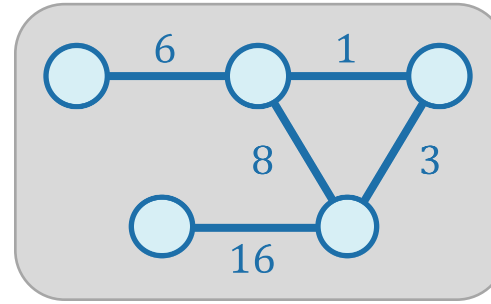
Sequential Local-Ratio



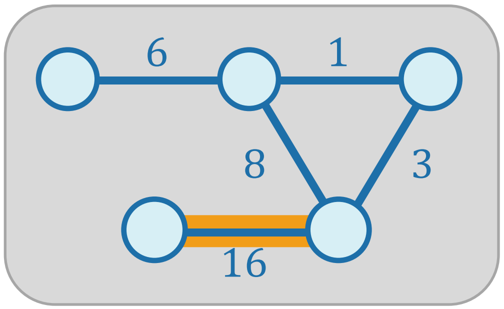




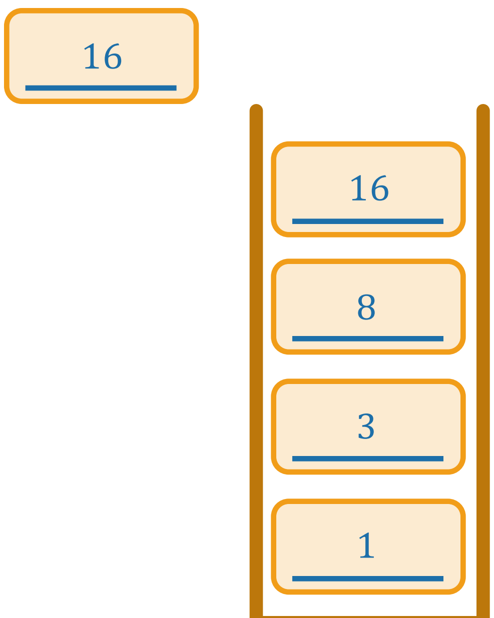
Done!
Start unwinding the stack



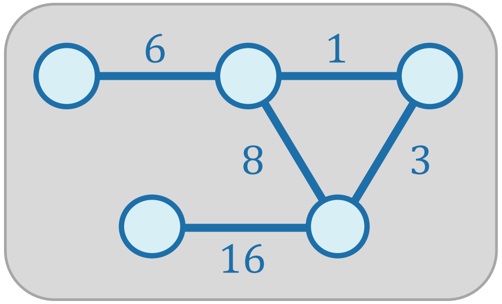
Sequential Local-Ratio

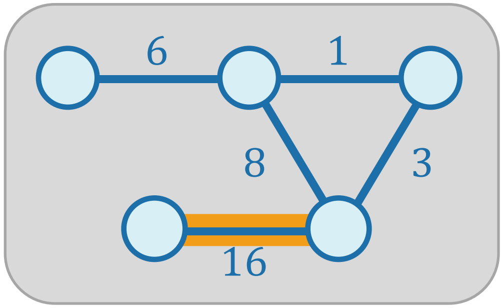


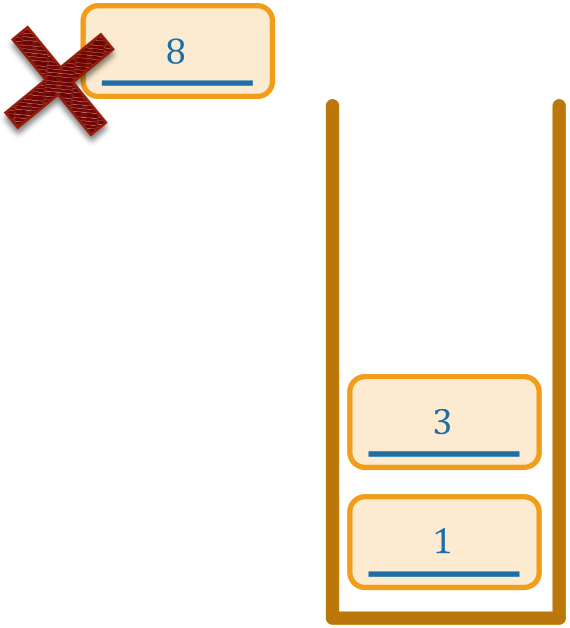
Greedy, by the stack order

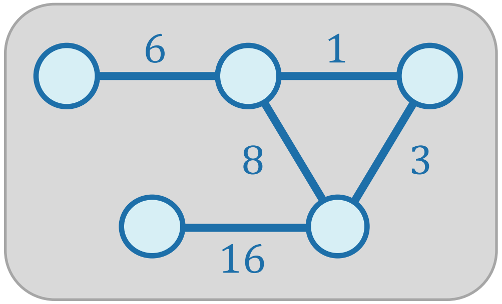


Sequential Local-Ratio

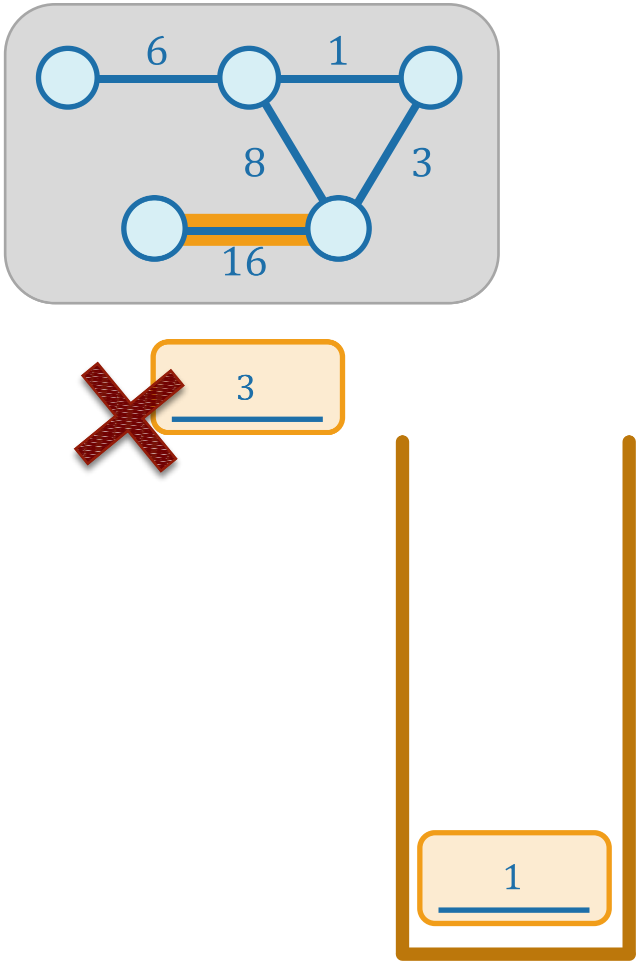


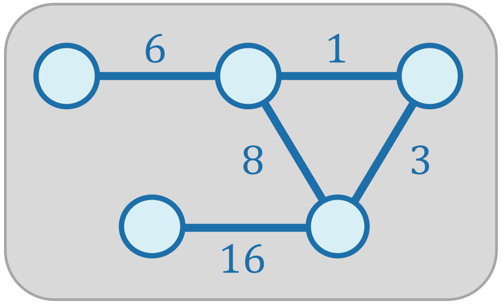




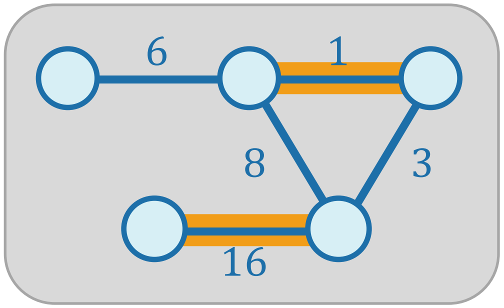


Sequential Local-Ratio





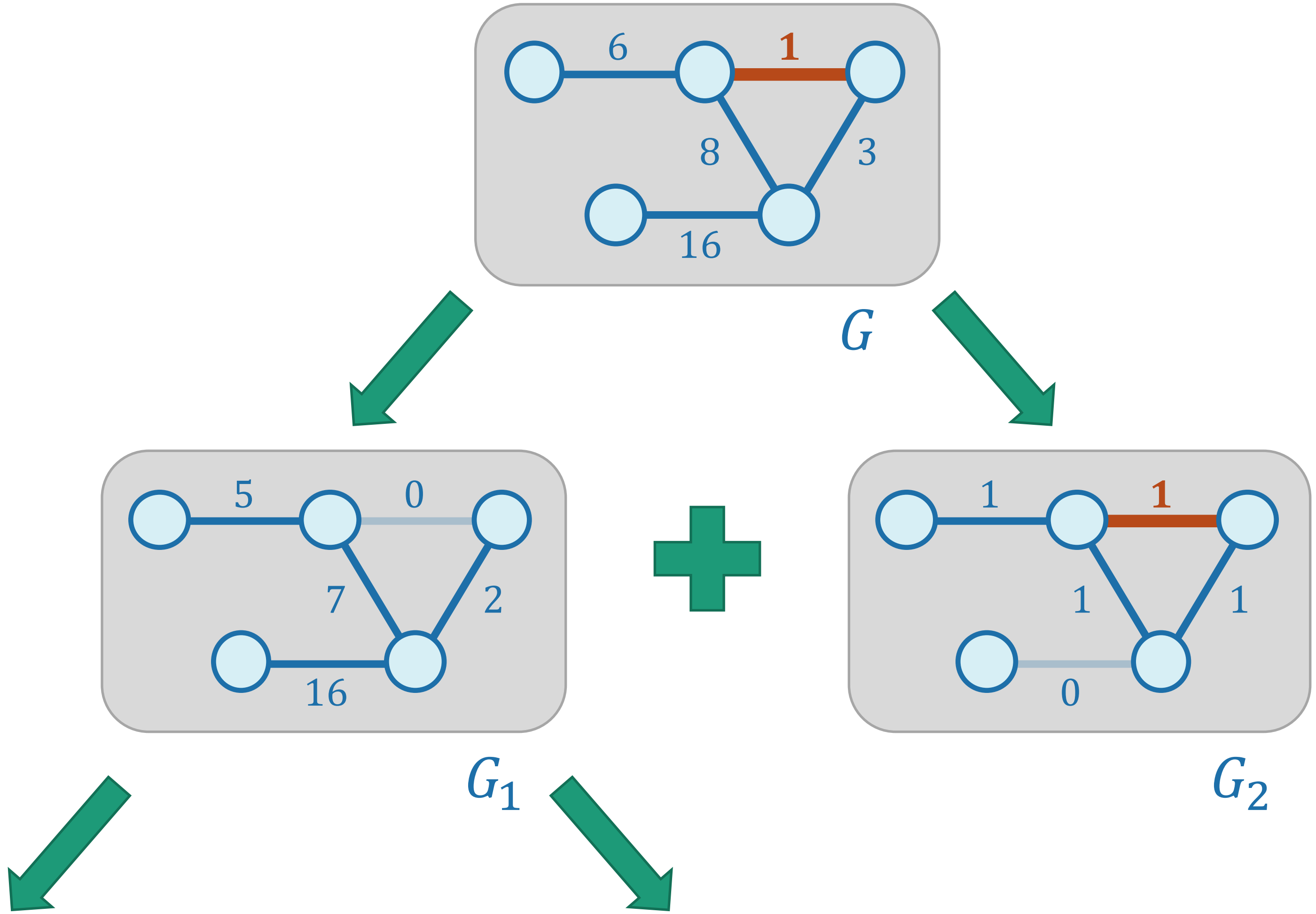
Sequential Local-Ratio



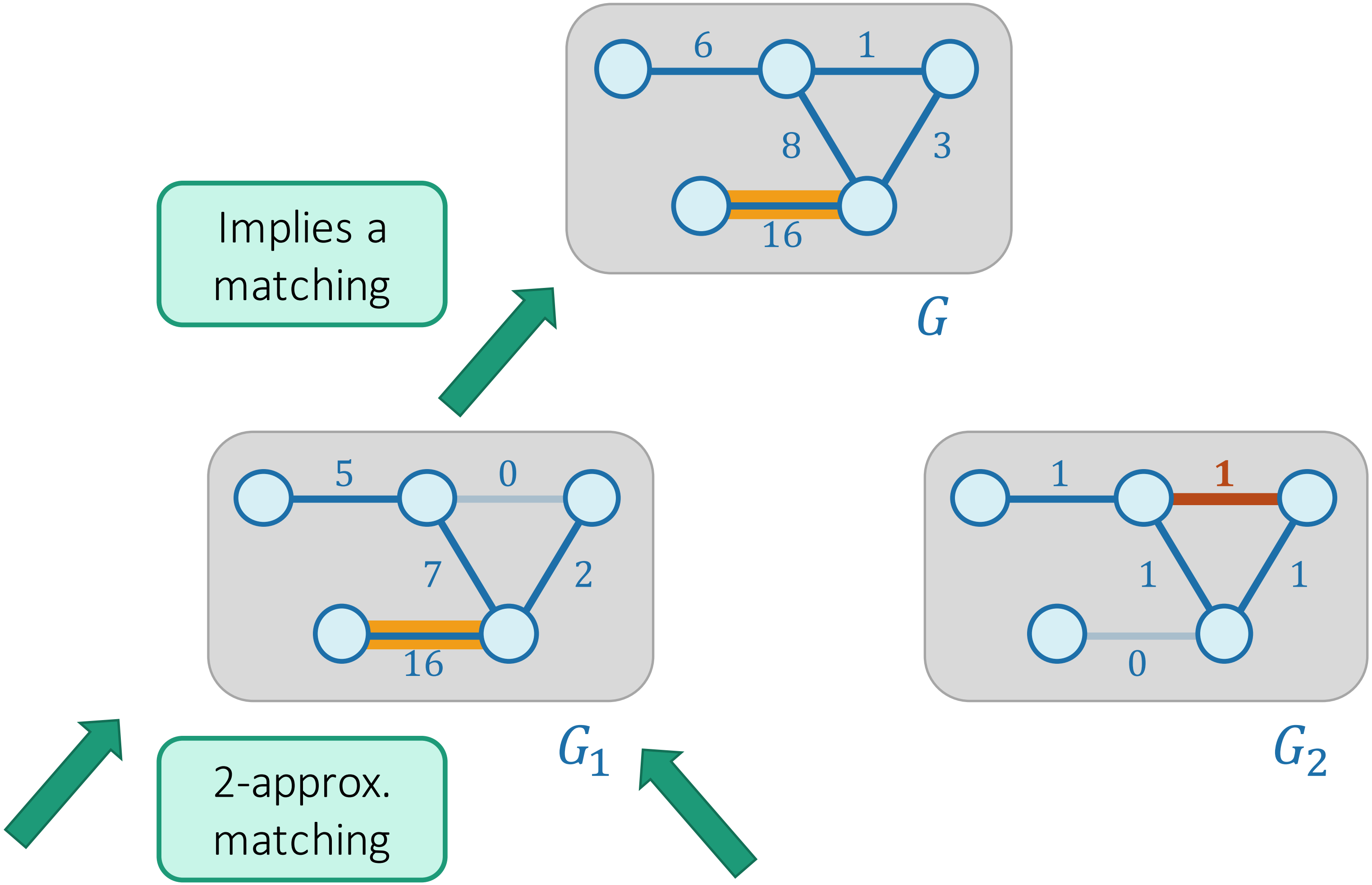
Output the matching

Alg: 17
Opt: 22

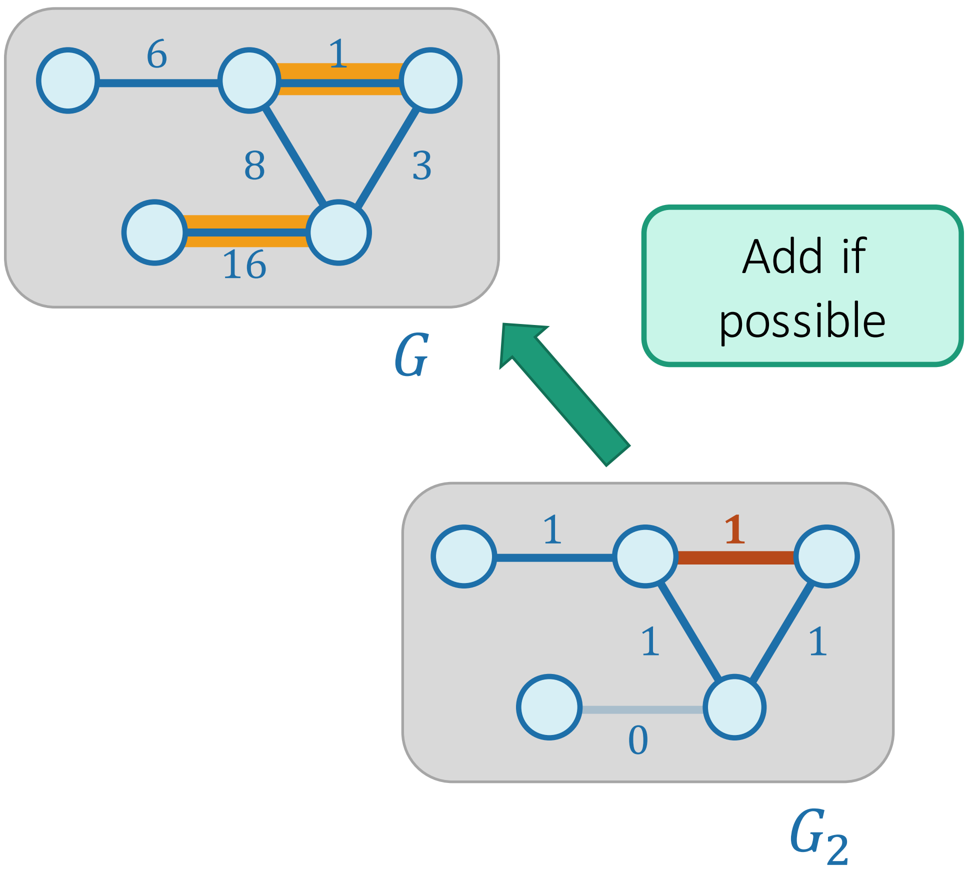
Analysis: a Single Step



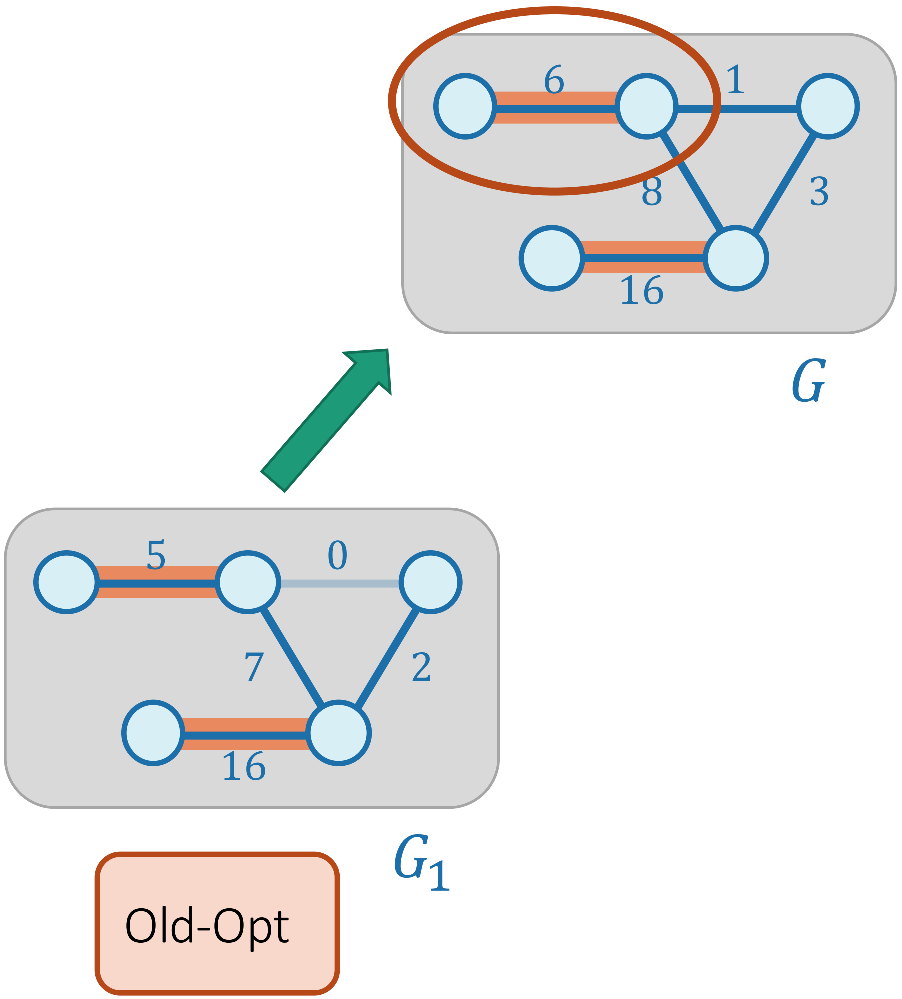
Analysis: a Single Step



Analysis: a Single Step



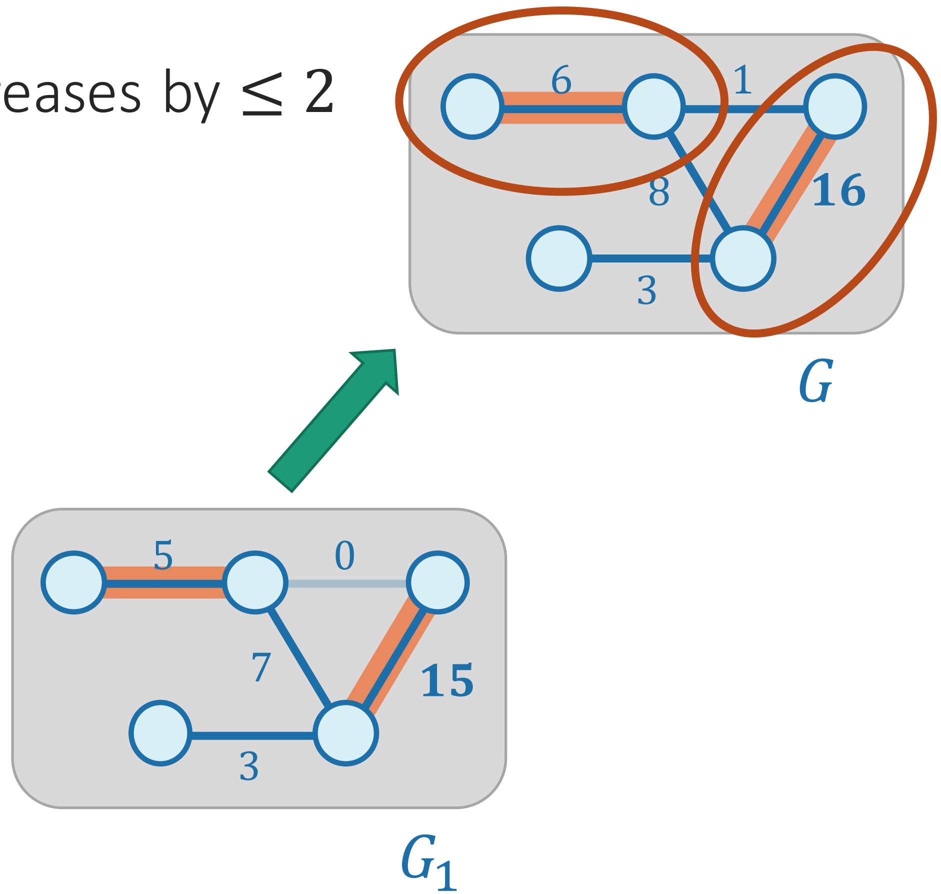
How does it Work?



Opt: +1

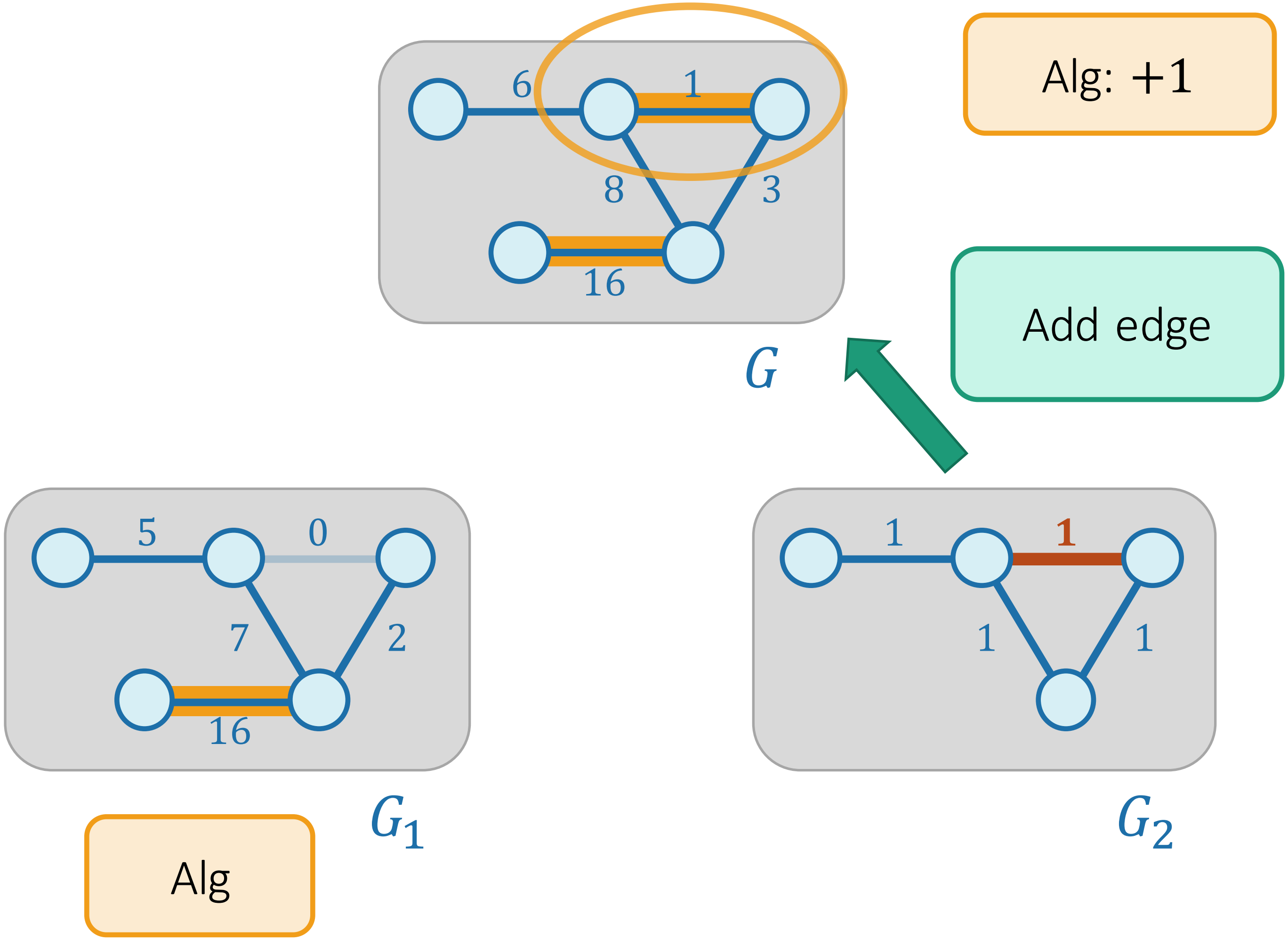
How does it Work?

Opt increases by ≤ 2



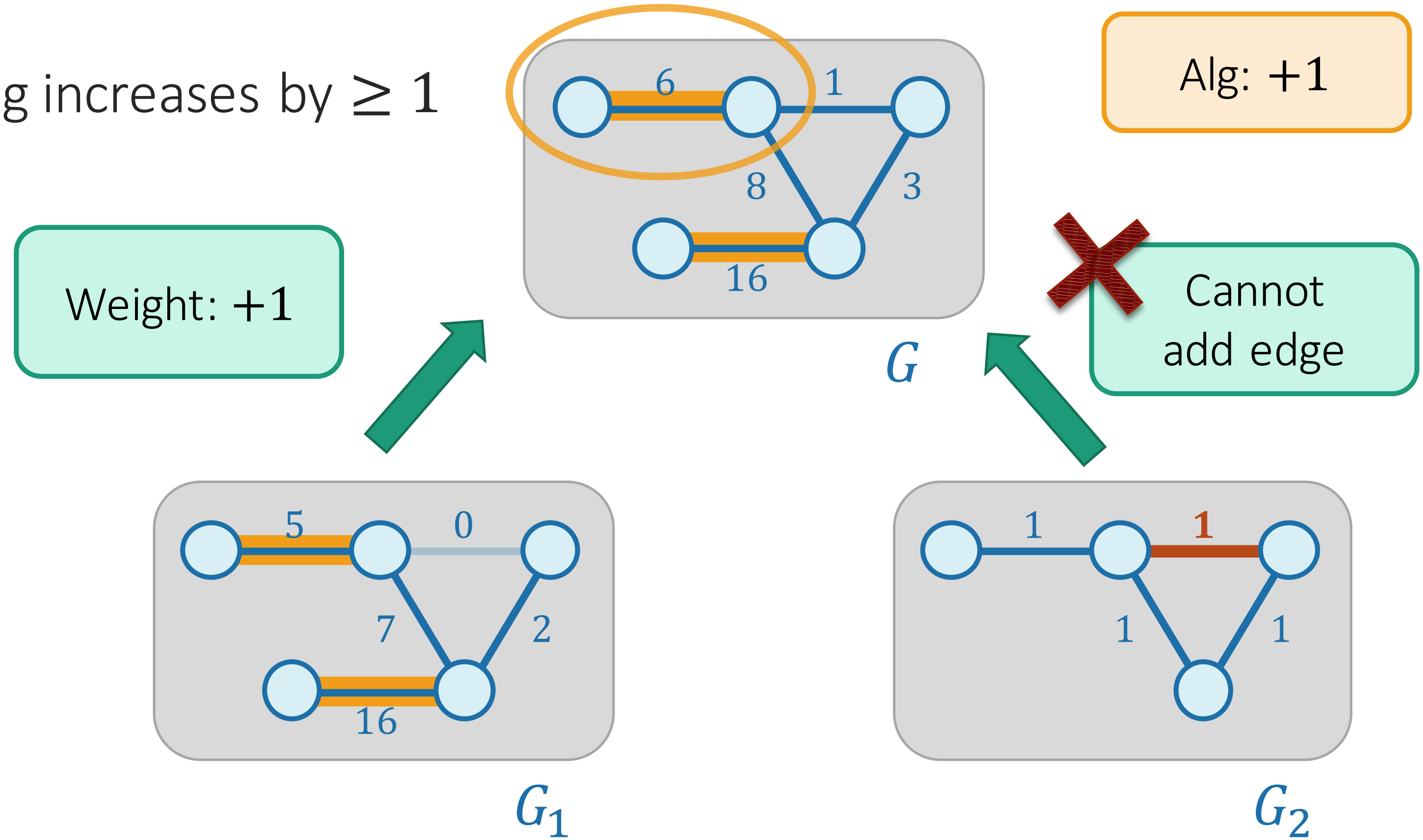
Opt: +2

How does it Work?



How does it Work?

Alg increases by ≥ 1



How does it Work?

Processing edge e:

- Opt increases by $\leq 2w(e)$
- Alg increases by $\geq 1w(e)$

2-approx. matching

Local-Ratio & Semi-Streaming

- Edge ordering?

Any order works!

Local-Ratio & Semi-Streaming

- Edge ordering ✓

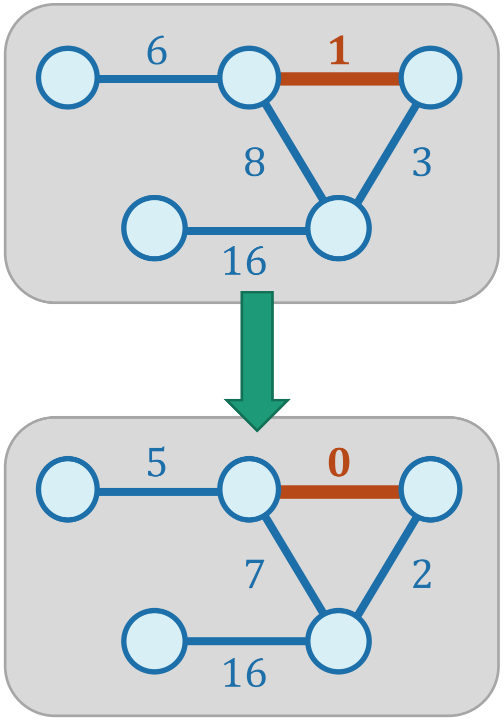
But…

- Reduce weights from future edges?

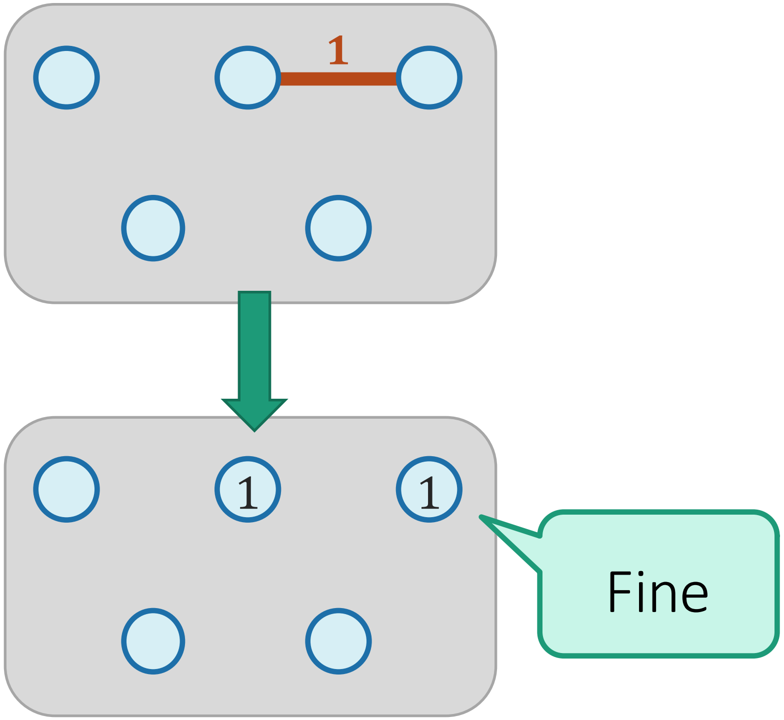
Retroactive reductions

Retroactive Reductions

Sequential:

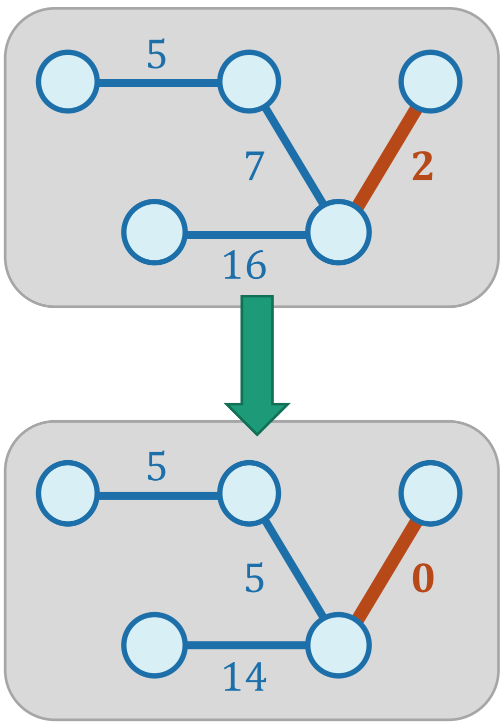


Streaming

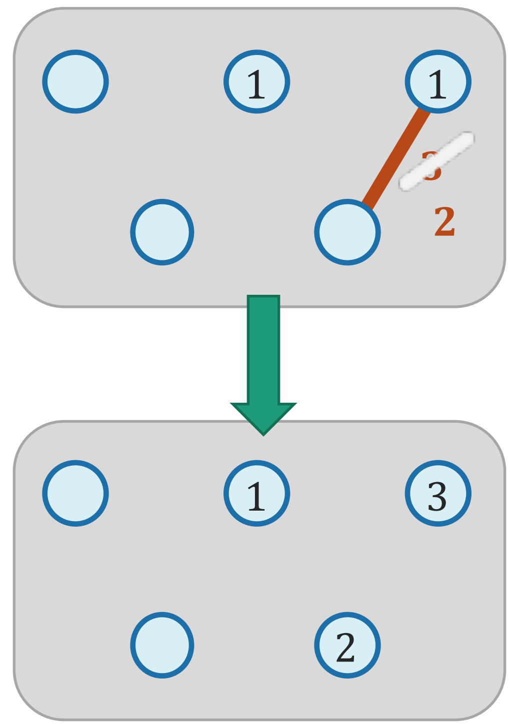


Retroactive Reductions

Sequential:



Streaming



Local-Ratio & Semi-Streaming

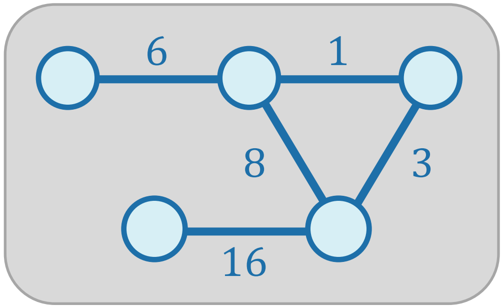
- Edge ordering ✓
- Retroactive reductions ✓

But…

- Memory usage?

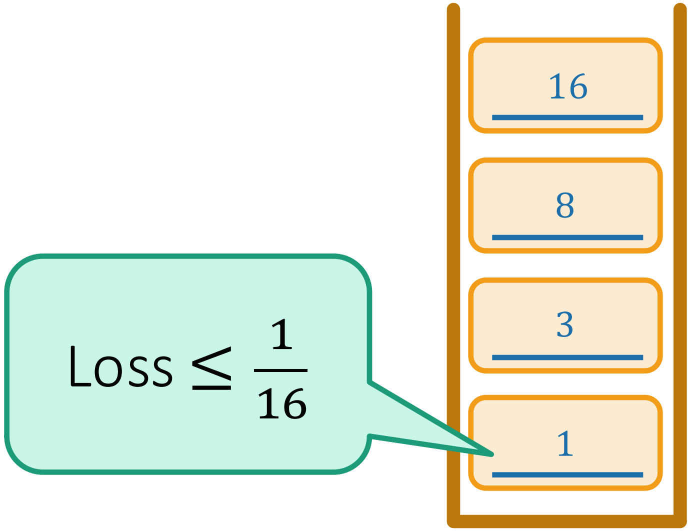
Remove light edges

Edge Removal



Stack:

- Alg $\geq$ maximum edge in stack
- Can remove light edges!



Local-Ratio & Semi-Streaming

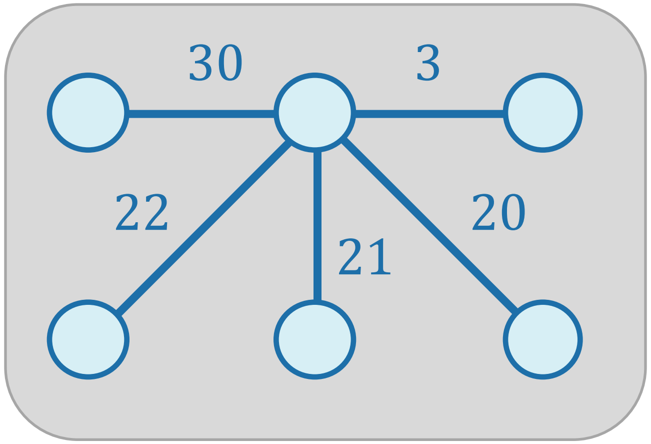
- Edge ordering ✓
- Retroactive reductions ✓
- Stack shrinking ✓

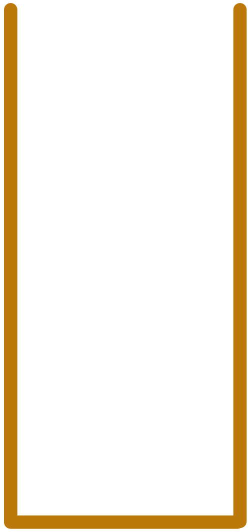
But how…

- Assure rapid edge-weight growth?

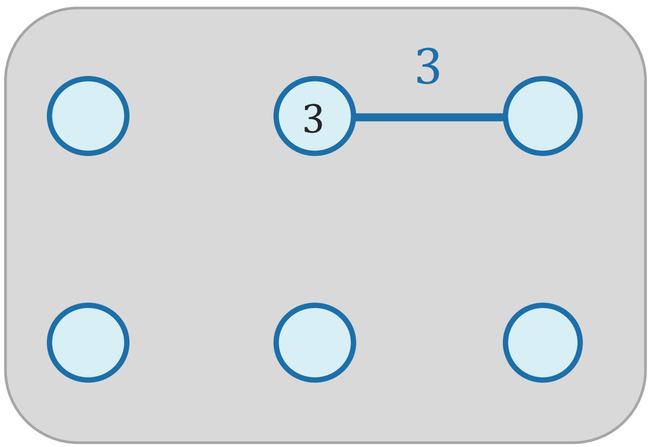
Asymmetric reductions

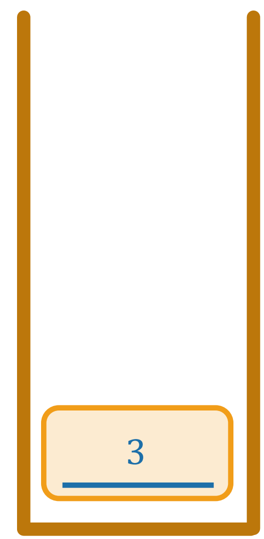
Example



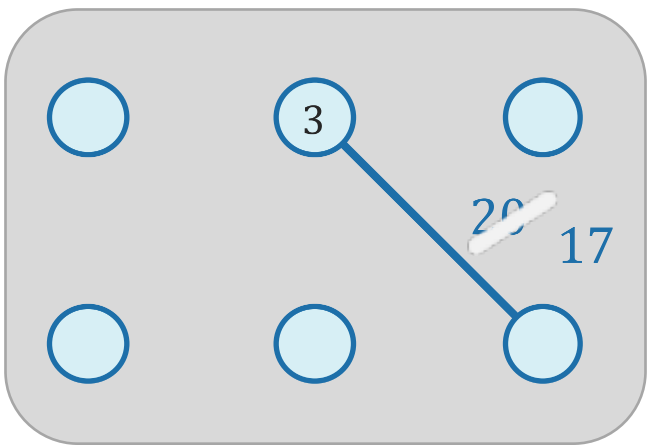


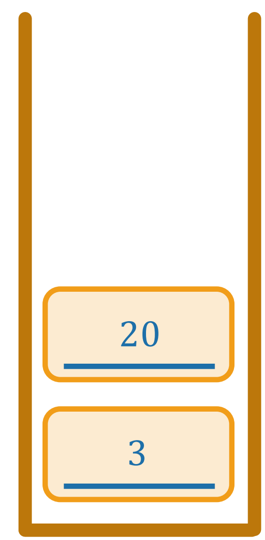
Example



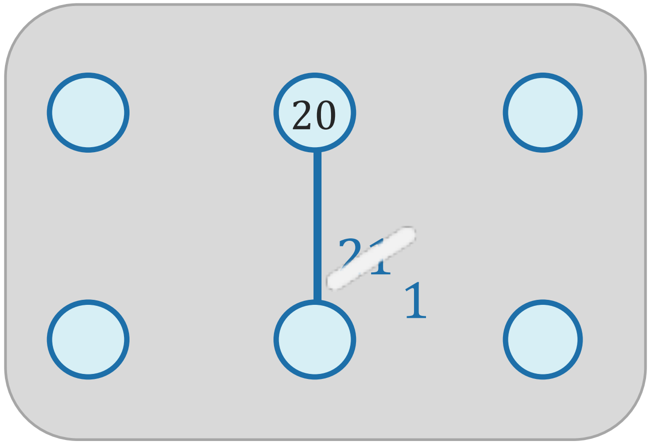


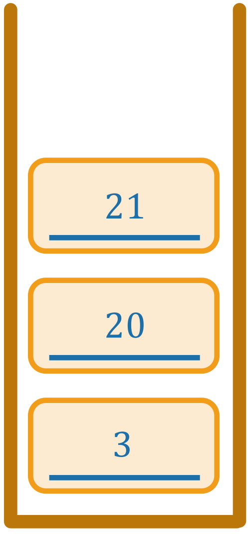
Example



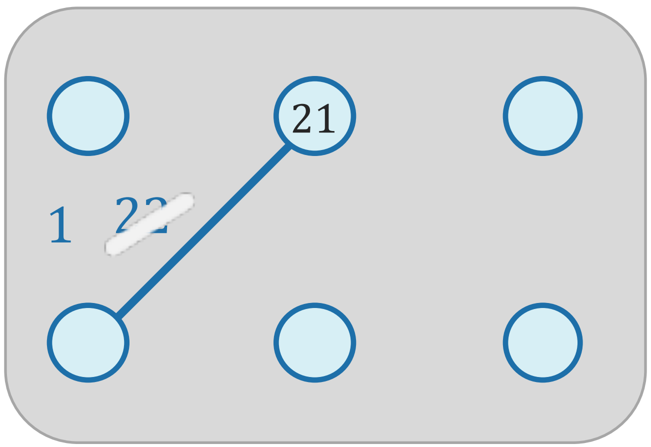


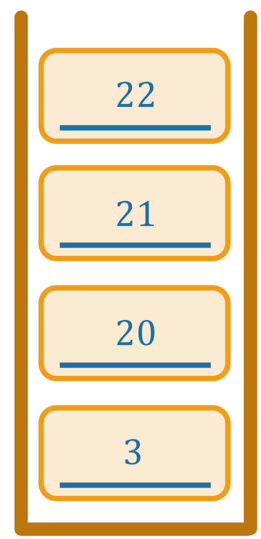
Example



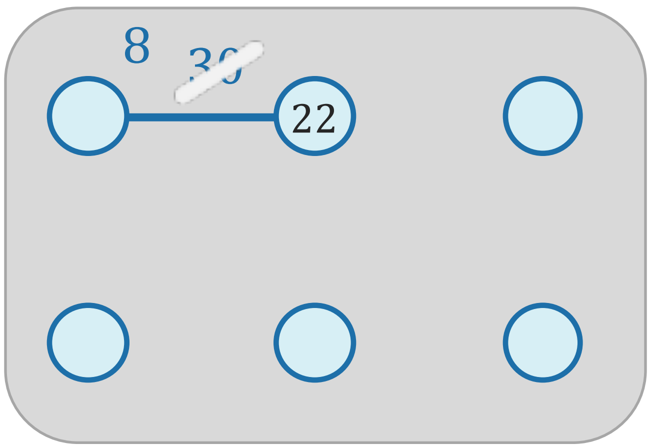


Example





Example



Stack overflow!!!

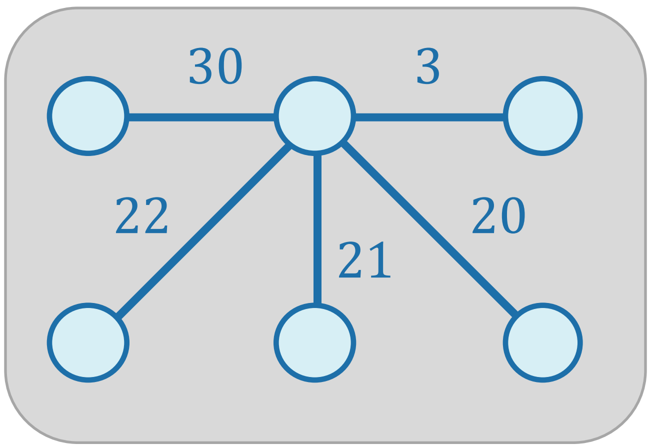
Asymmetric Reductions

Reduce $\alpha w(e)$ from some of the adjacent edges

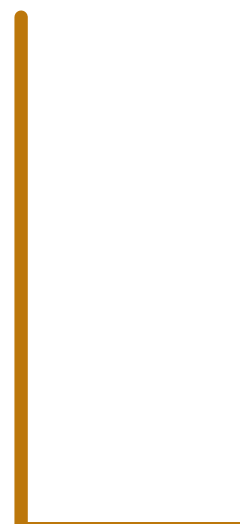
- $\alpha = (1 + \epsilon)$

2α-approx. matching

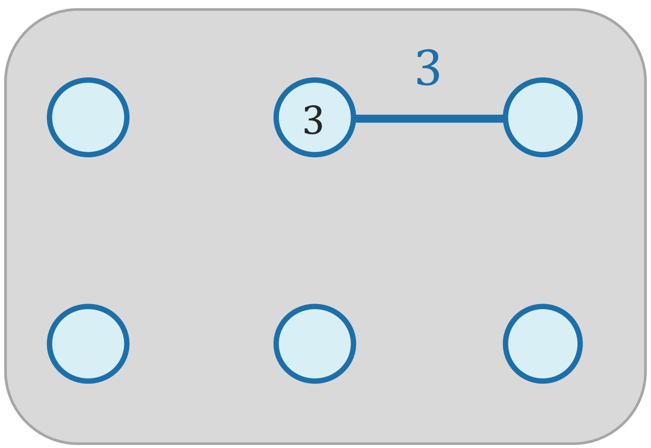
Asymmetric Reductions – Example

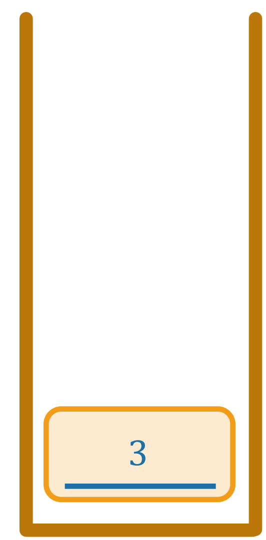


Again, with
asymmetric reductions

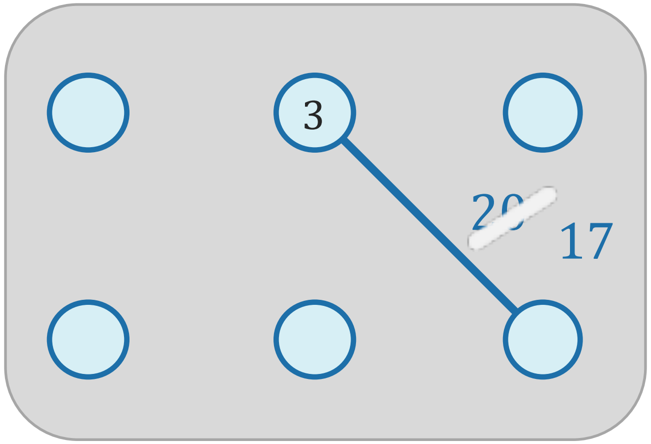


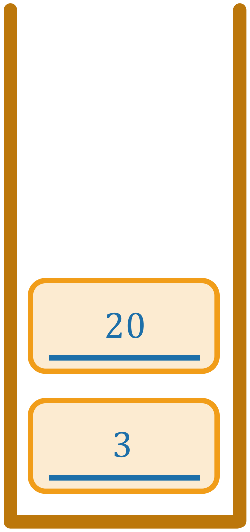
Asymmetric Reductions – Example



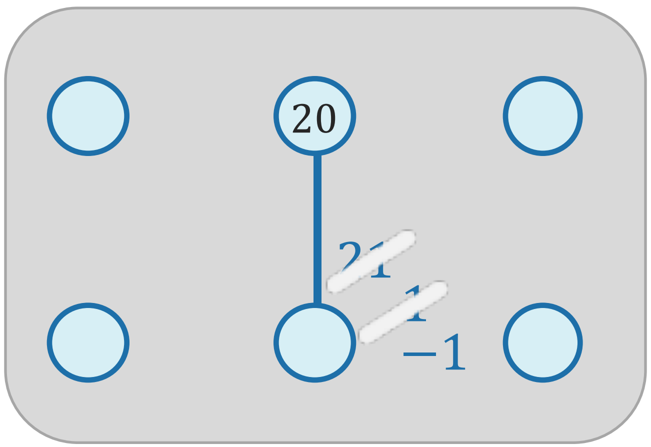


Asymmetric Reductions – Example

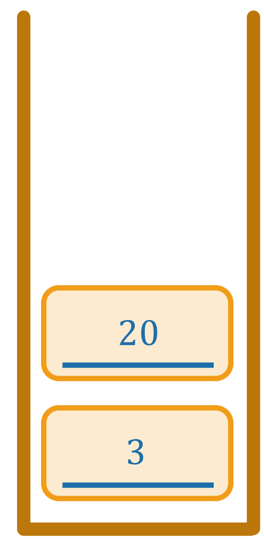




Asymmetric Reductions – Example

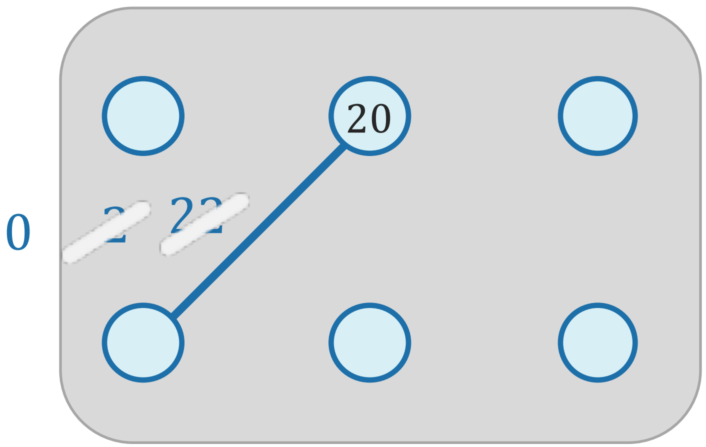


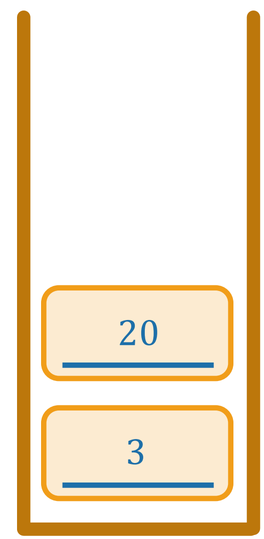
$1 \ll 20$ so discard it



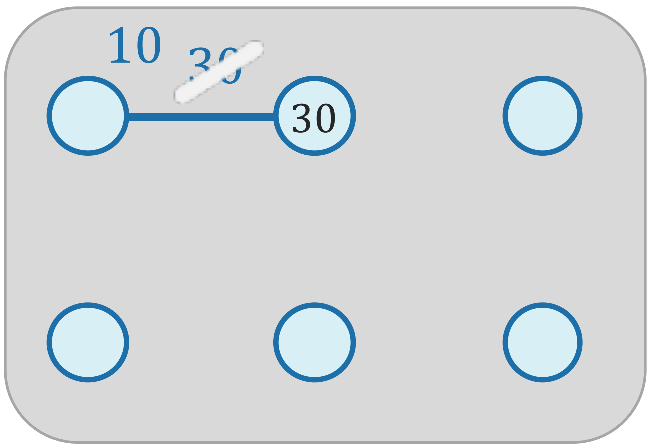
Asymmetric Reductions – Example

$2 \ll 20$ so discard it



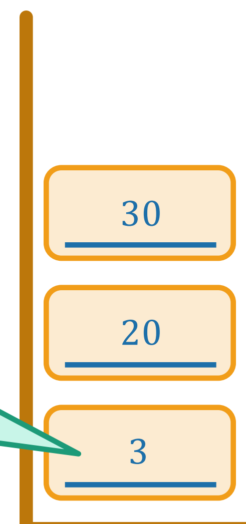


Asymmetric Reductions – Example

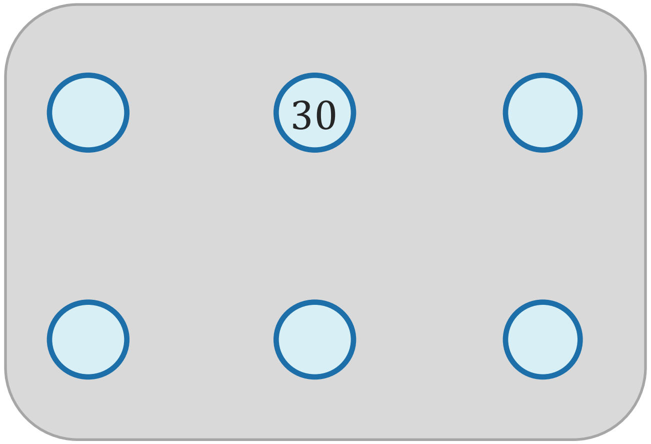


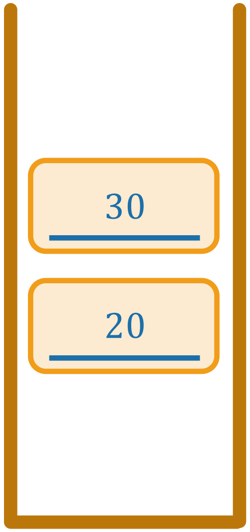
Remove light edges

Loss $\leq 10\%$



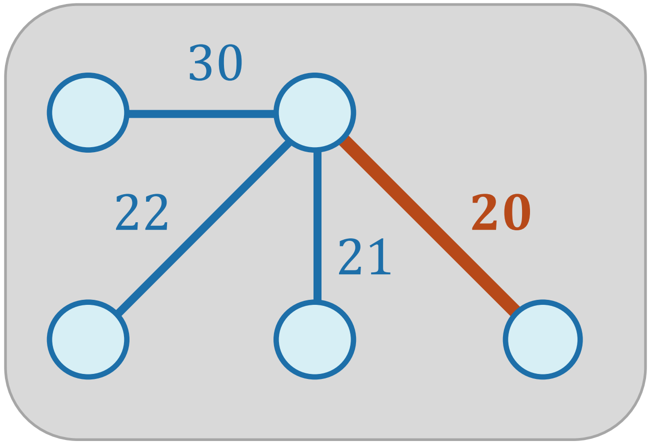
Asymmetric Reductions – Example



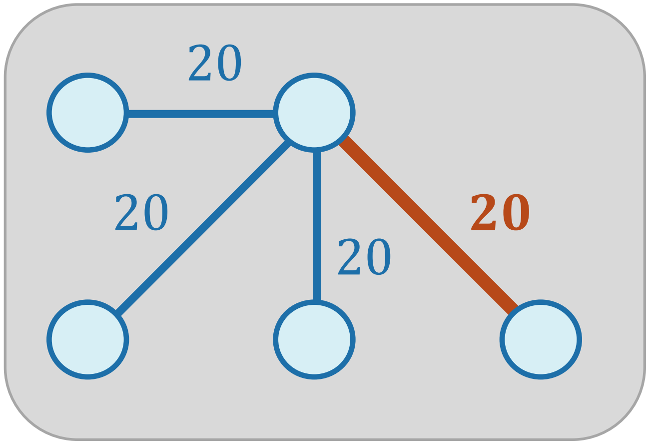


Smaller stack,
similar approx. ratio

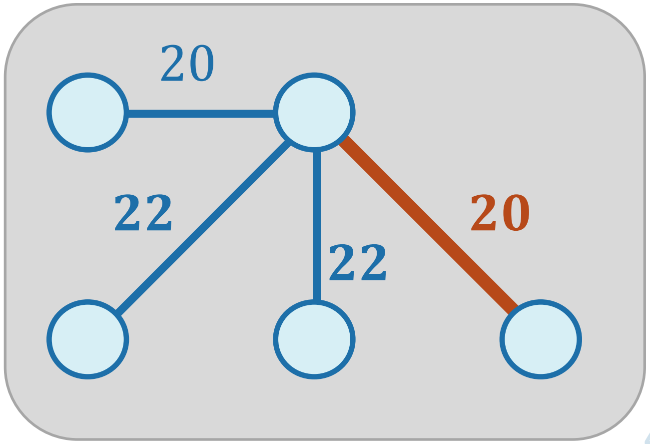
Example – Analysis



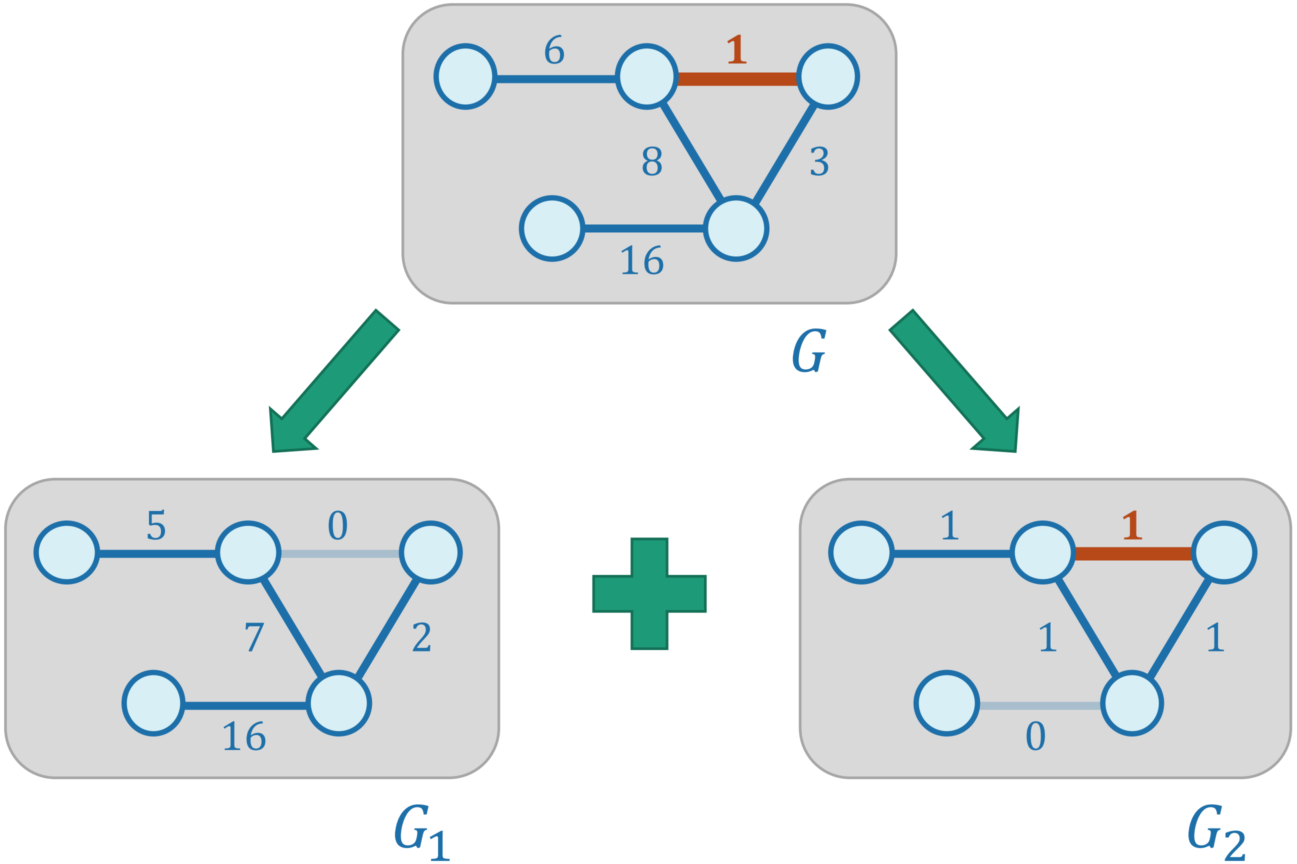
Original reduction:



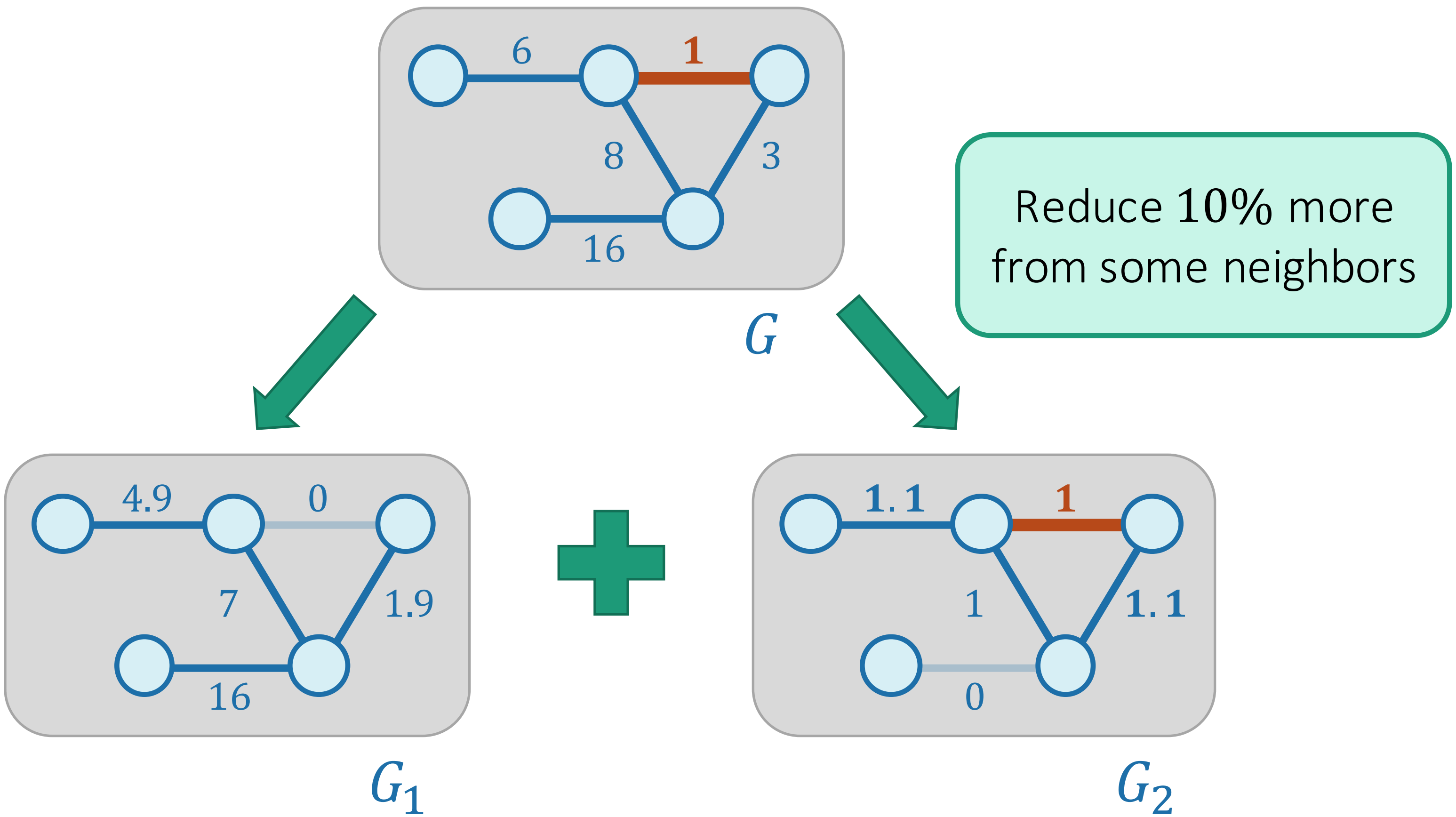
New reduction:



Correctness, Revised



Correctness, Revised



Correctness, Revised

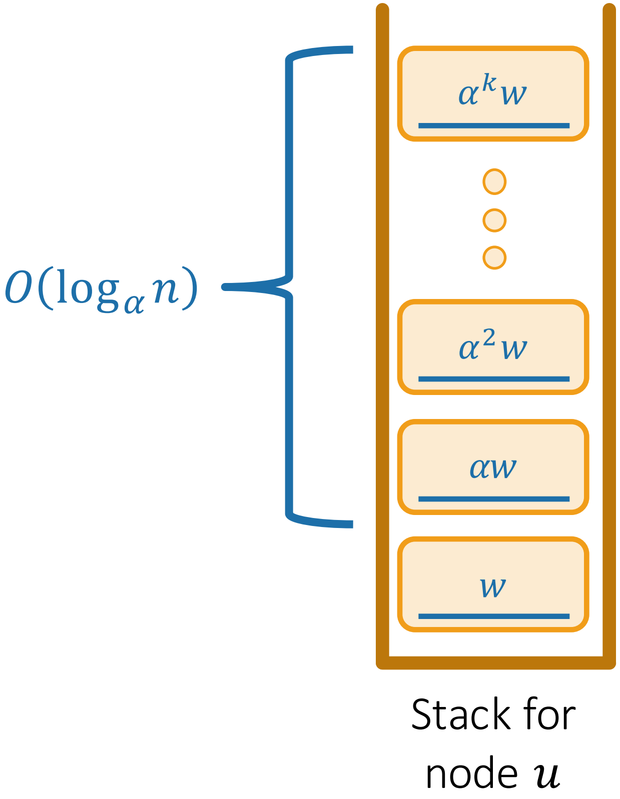
Processing edge e:

- Opt increases by $\leq 2.2w(e)$
- Alg increases by $\geq 1w(e)$

2.2-approx. matching

Stack Size

- If $w(u,v) \leq \alpha\big(f(u) + f(v)\big)$, ignore
- Otherwise, reduce $f(u) + f(v)$
- Throw away light edges



Local-Ratio & Semi-Streaming

- Edge ordering ✓
- Retroactive reductions ✓
- Stack shrinking ✓
- Asymmetric reductions ✓

Complexity

- Memory usage:
  - Per node: $O(\log n)$ edges
  - Total: $\boldsymbol{O(n\log^2 n)}$ **bits**
- Time complexity:
  - Per edge: $O(1)$ time
  - Post-processing: $O(n\log n)$ time
- Deterministic

In Conclusion

- $(2+\epsilon)$-approx. for MWM in semi-streaming
- Local-ratio + semi-streaming = ♥
- Technical contributions:
  - Retroactive reductions
  - Stack shrinking
  - Asymmetric reductions
- Other applications for local-ratio?

Thank You!