Distributed Derandomization via Network Decomposition

•••

Václav Rozhoň (ETH)

joint work with
Mohsen Ghaffari (ETH)

We will see...

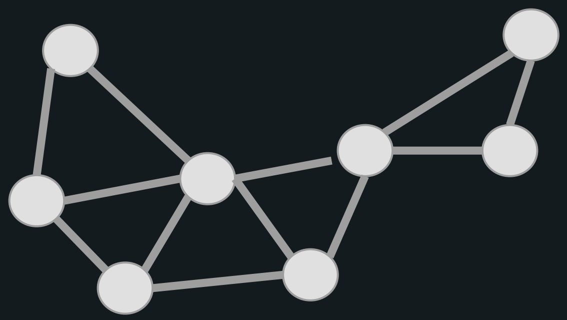
...distributed deterministic algorithm for **network decomposition.**

This is a key technical tool for a lot of theory built in past few years.

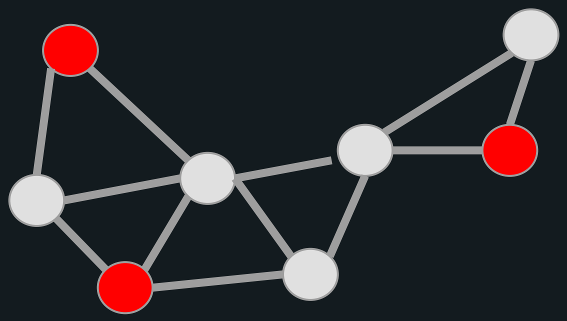
Plan:

1. Define the distributed LOCAL model
2. Survey implications of the algorithm
3. Give definition of net. decomposition and algorithm for it

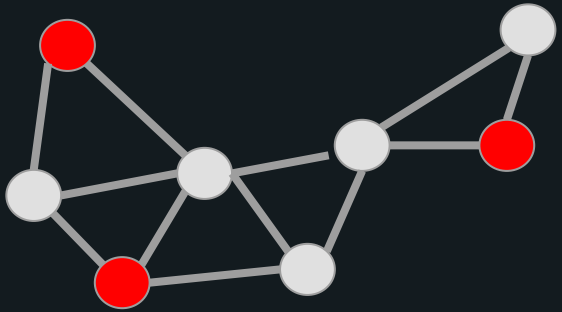
Maximal independent set

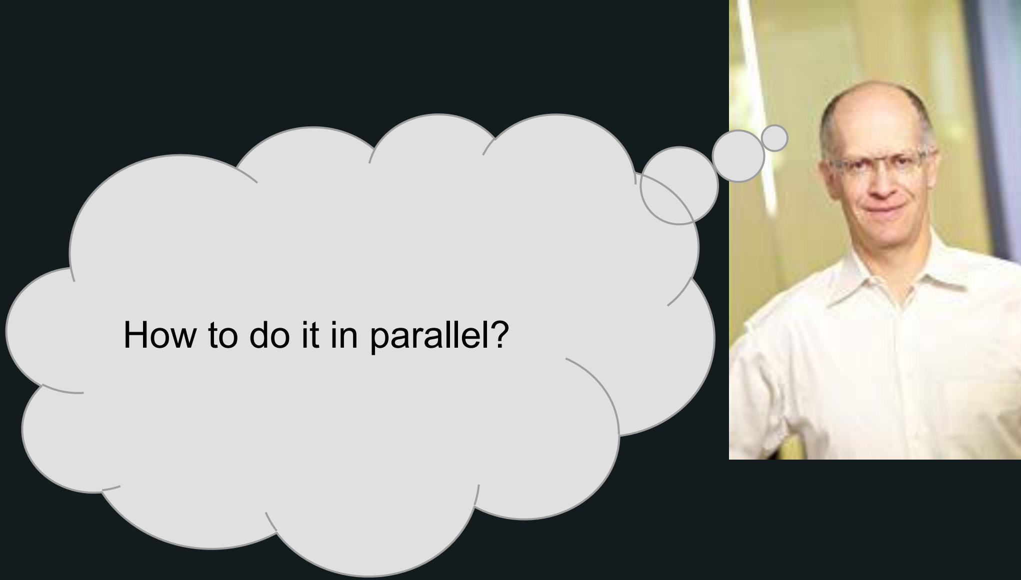


Maximal independent set

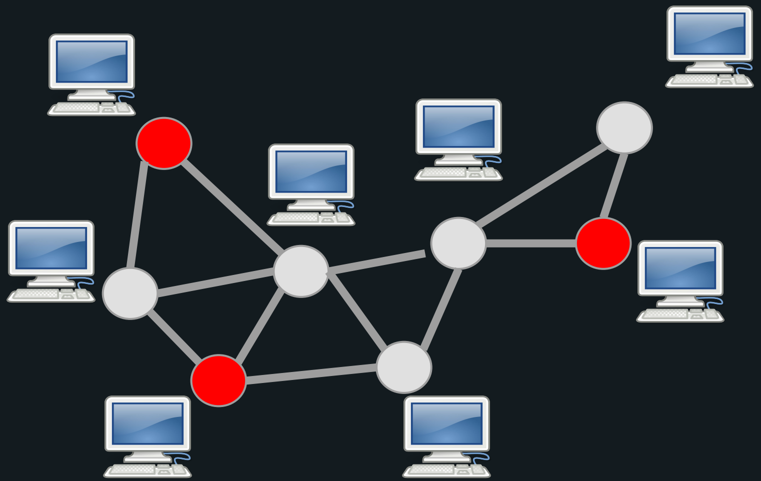


Maximal independent set





Maximal independent set

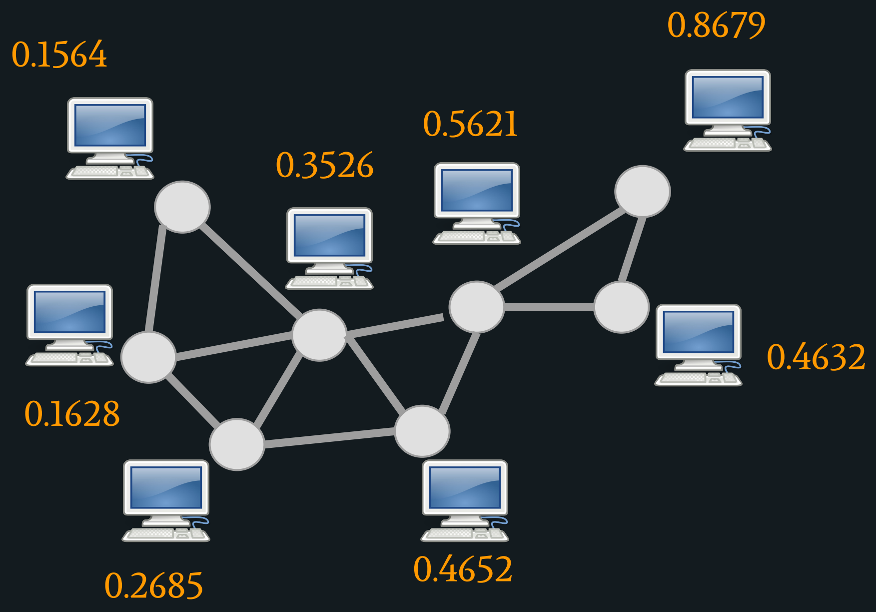


How to do it in parallel?

Luby's algorithm

[Luby STOC'85; Alon, Babai, Itai JoA'86]

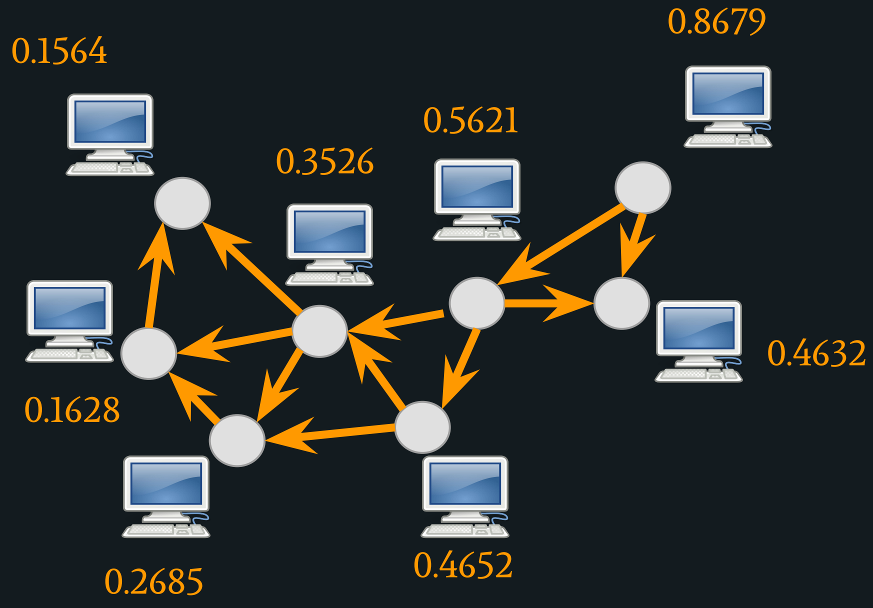
Maximal independent set



Each node samples a random number from [0,1]

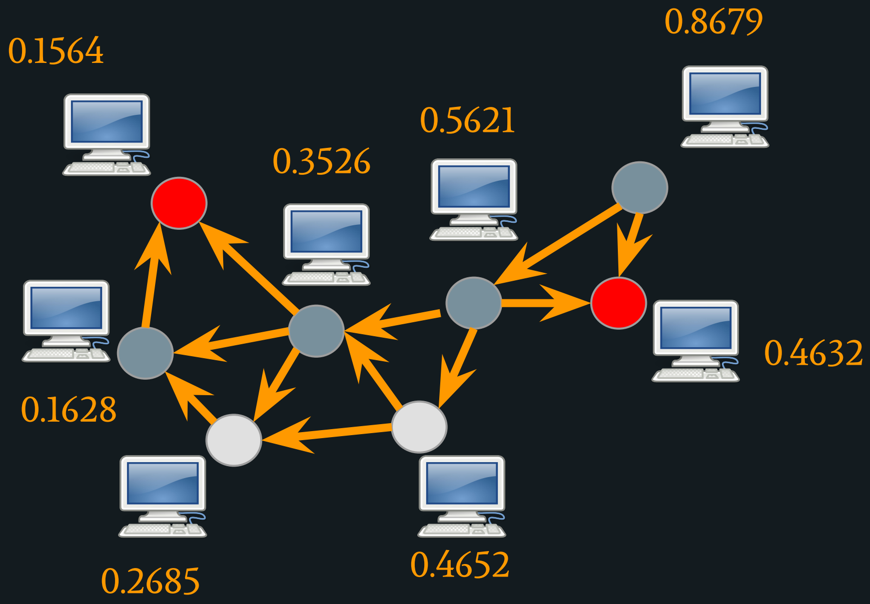
Maximal independent set

Luby's algorithm [Luby STOC'85]



Maximal independent set

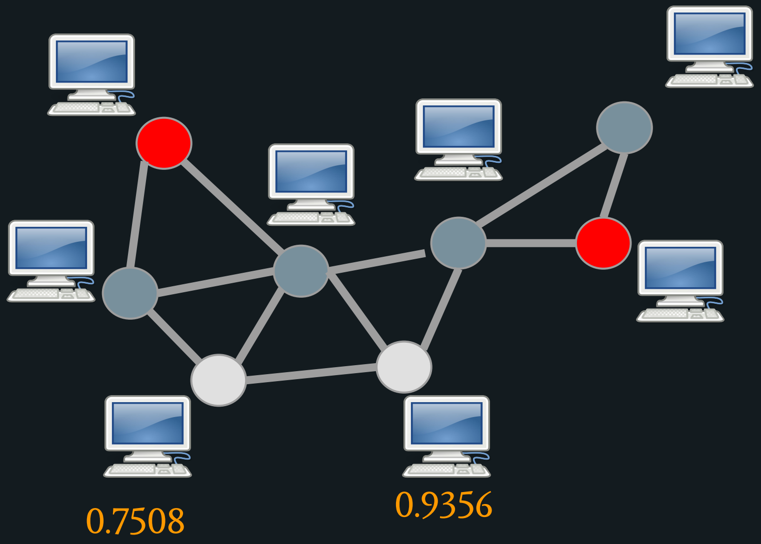
Luby’s algorithm [Luby STOC’85]



Let’s add them to current solution.

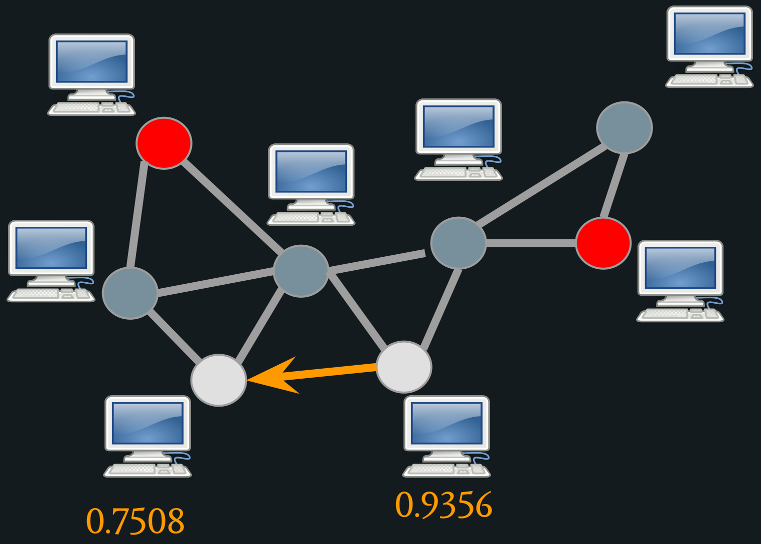
Maximal independent set

Luby's algorithm [Luby STOC'85]



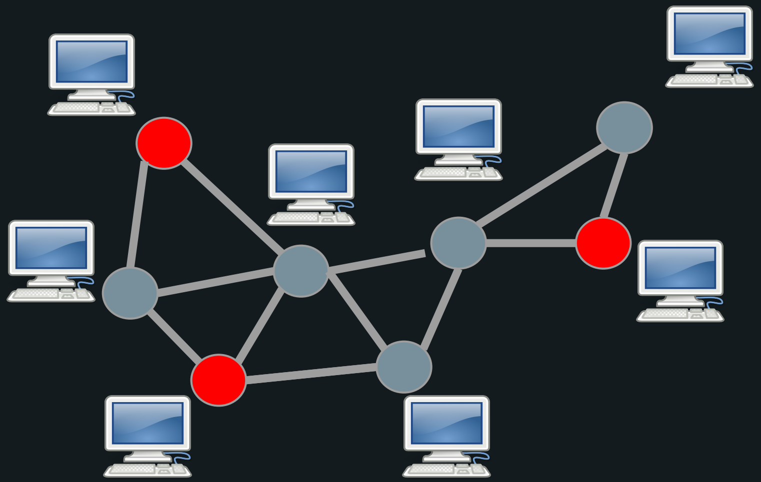
Maximal independent set

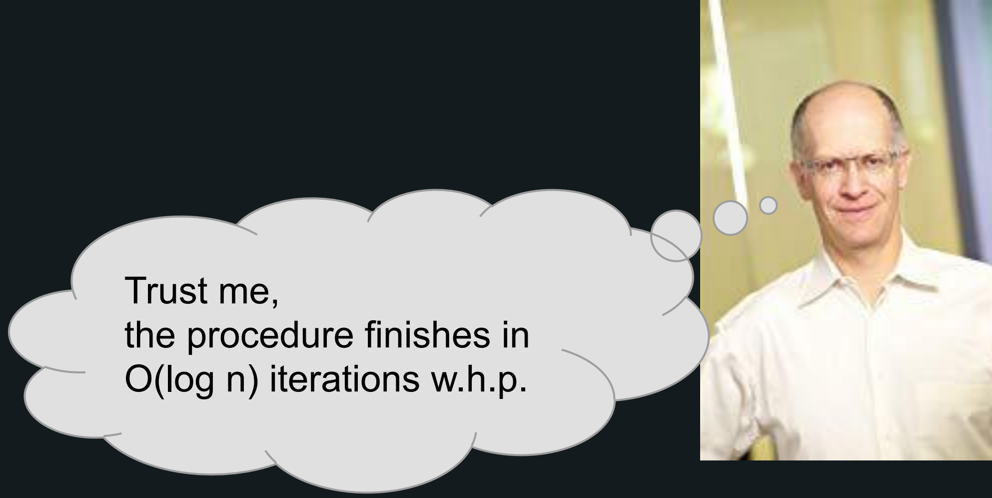
Luby's algorithm [Luby STOC'85]



Maximal independent set

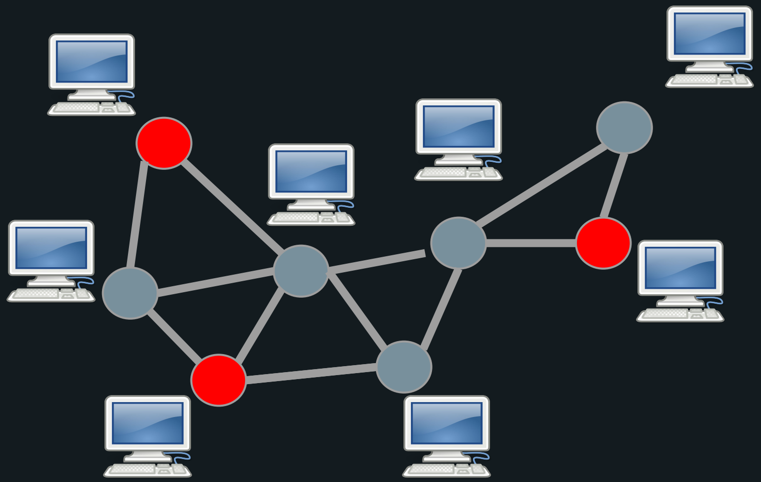
Luby's algorithm [Luby STOC'85]

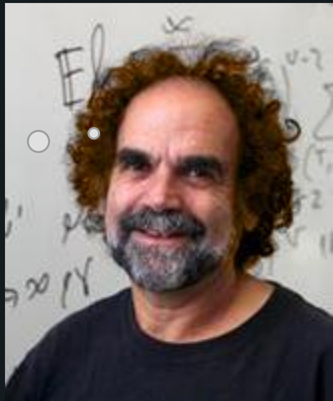




Maximal independent set

Luby's algorithm [Luby STOC'85]





Let me tell you now what the model is.

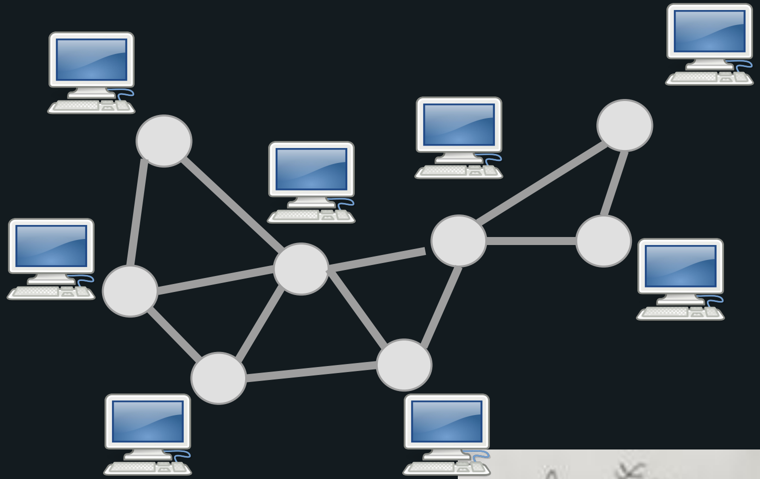
The LOCAL model of distributed graph algorithms

- Undirected graph $G=(V,E)$ with n nodes
- One computer in each node

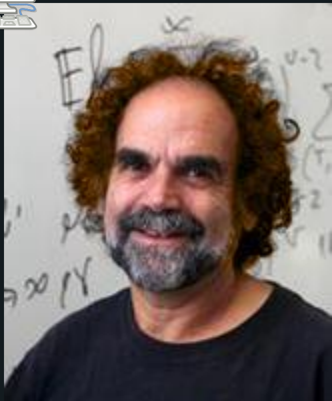
- Synchronous message passing rounds
- Unbounded message size and computation! (more honest version: CONGEST model)

- Initially, nodes know only (upper bound on) n and their unique $O(\log n)$ bit label
- In the end, each node should know its part of output

- Time complexity: number of rounds



LOCAL model
[Linial FOCS'87]



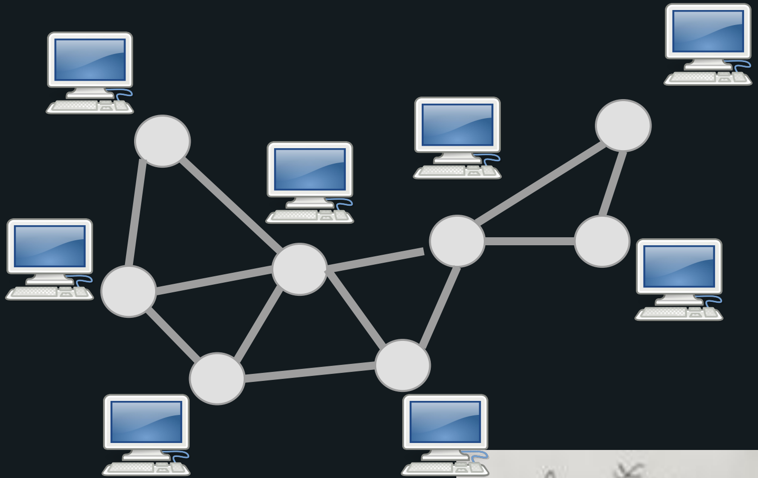
The LOCAL model of distributed graph algorithms

- Undirected graph *G=(V,E)* with *n* nodes
- One computer in each node

- Synchronous message passing rounds
- Unbounded message size and computation! (more honest version: CONGEST model)

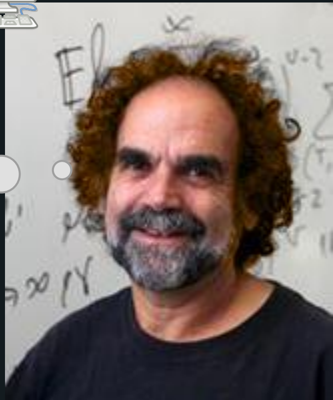
- Initially, nodes know only (upper bound on) *n* and their unique *O(*log *n)* bit label
- In the end, each node should know its p

- Time complexity: number of rounds



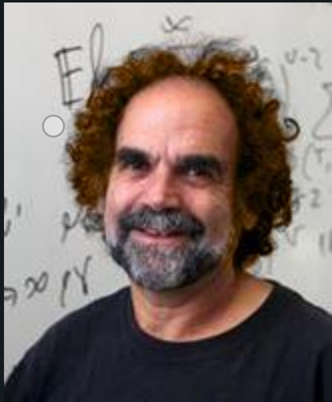
odel
S'87]

We neglect:
computation,
congestion,
asynchronicity,
fault-tolerance, …



Deterministic maximal independent set

Is there also an efficient (polylogarithmic round) deterministic algorithm for MIS? [Linial FOCS'87]



Deterministic maximal independent set

Yes, it directly follows from our algorithm for **network decomposition.** [R., Ghaffari 19+]

Also: $\Delta+1$ coloring, maximal matching, Lovasz Local Lemma, hypergraph splitting,...

Deterministic maximal independent set

Yes, it directly follows from our algorithm for **network decomposition.** [R., Ghaffari 19+]

Also: $\Delta+1$ coloring, maximal matching, Lovasz Local Lemma, hypergraph splitting,...

Wait a minute!
[Fischer DISC'17]

Derandomization

[Ghaffari, Kuhn, Maus STOC'17] + [Ghaffari, Harris, Kuhn FOCS'18] + [R., Ghaffari '19+]:

"For all problems that allow polylogarithmic-round randomized algorithm*, there is also a polylogarithmic-round **deterministic** algorithm. "

*whose solution can be checked deterministically in polylogarithmic number of rounds

Randomized algorithms

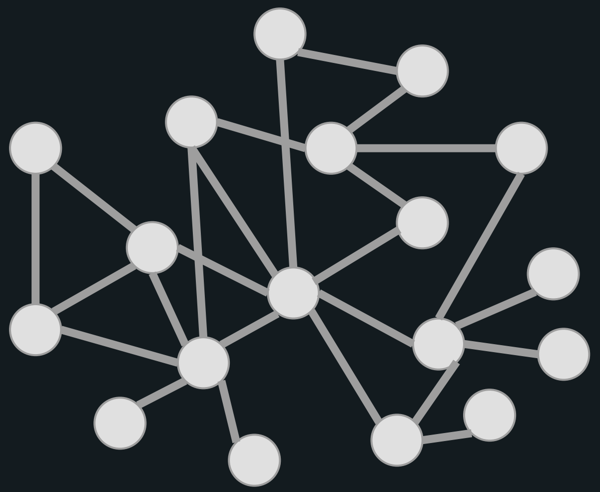
	deterministic	randomized
Network decomposition	$2^{O(\sqrt{\log n \log\log n})}$ [Awerbuch, Goldberg, Luby, Plotkin FOCS'89] $2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC'92] $O(\log^7 n)$[R., Ghaffari '19+]	$O(\log n)$ [Linial, Saks SODA'91]
Δ+1 coloring	$2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC'92] $O(\log^7 n)$[R., Ghaffari '19+]	$2^{O(\sqrt{\log\log n})}$ [Chang, Li, Pettie STOC'18] $O(\text{poly}(\log\log n))$ [R., Ghaffari '19+]
maximal independent set (MIS)	$2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC'92] $O(\log^7 n)$[R., Ghaffari '19+]	$O(\log\Delta) + 2^{O(\sqrt{\log\log n})}$ [Ghaffari SODA'16] $O(\log\Delta + \text{poly}(\log\log n))$ [R., Ghaffari '19+]

Randomized algorithr

[Chang, Kopelowitz, Pettie FOCS'16]:
Improvement of the deterministic side is actually necessary!

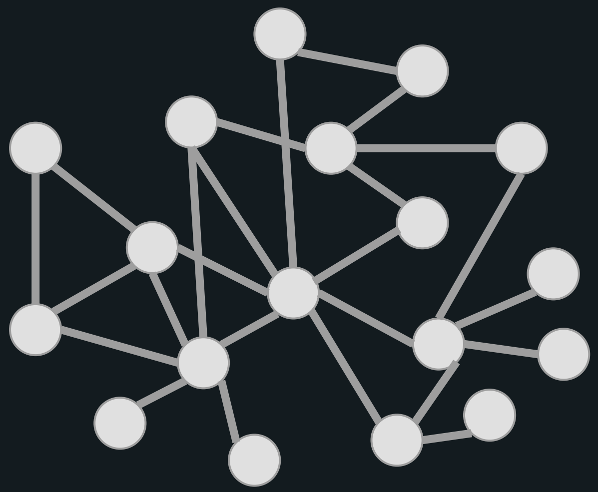
	deterministic	
Network decomposition	$2^{O(\sqrt{\log n \log\log n})}$ [Awerbuc $2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC $O(\log^7 n)$[R., Ghaffari '19+]	$\log n)$ [Linial, Saks SODA'91]
Δ+1 coloring	$2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC'92] $O(\log^7 n)$[R., Ghaffari '19+]	$2^{O(\sqrt{\log\log n})}$ [Chang, Li, Pettie STOC'18] $O(\mathrm{poly}(\log\log n))$ [R., Ghaffari '19+]
maximal independent set (MIS)	$2^{O(\sqrt{\log n})}$ [Panconesi, Srinivasan STOC'92] $O(\log^7 n)$[R., Ghaffari '19+]	$O(\log\Delta) + 2^{O(\sqrt{\log\log n})}$ [Ghaffari SODA'16] $O(\log\Delta + \mathrm{poly}(\log\log n))$ [R., Ghaffari '19+]

Principled approach for MIS



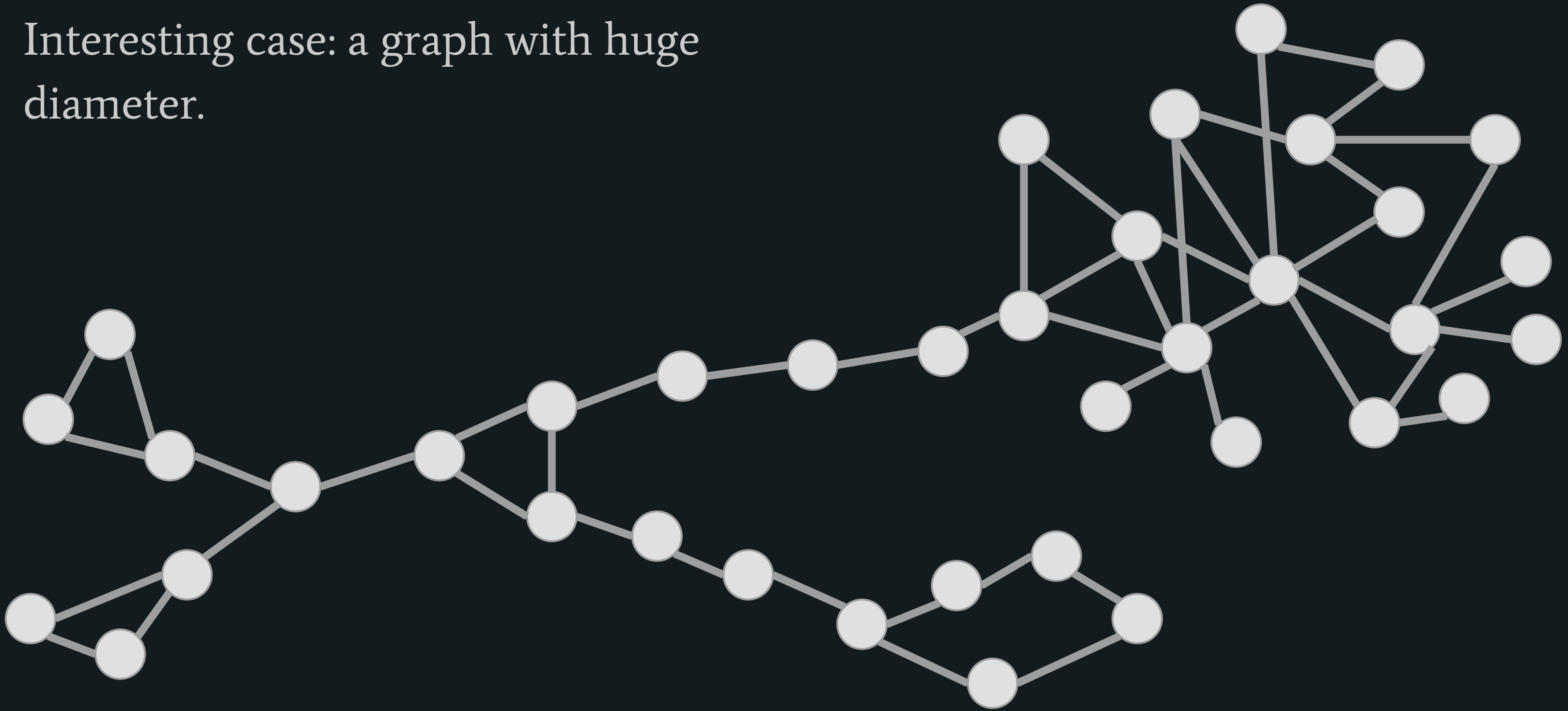
Principled approach for MIS

Easy case for our quest: the underlying graph has small (polylogarithmic) diameter.

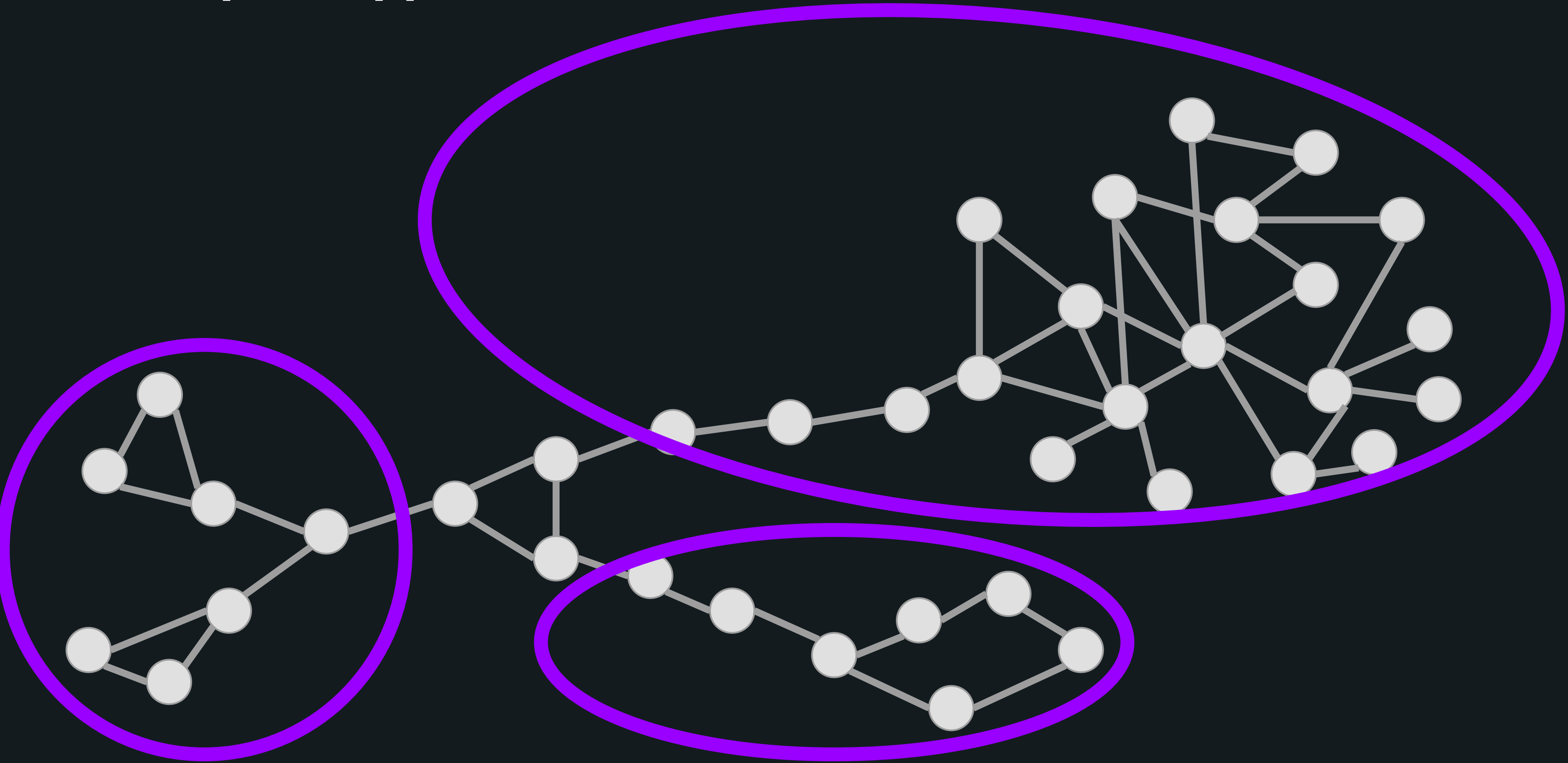


Principled approach

Interesting case: a graph with huge diameter.

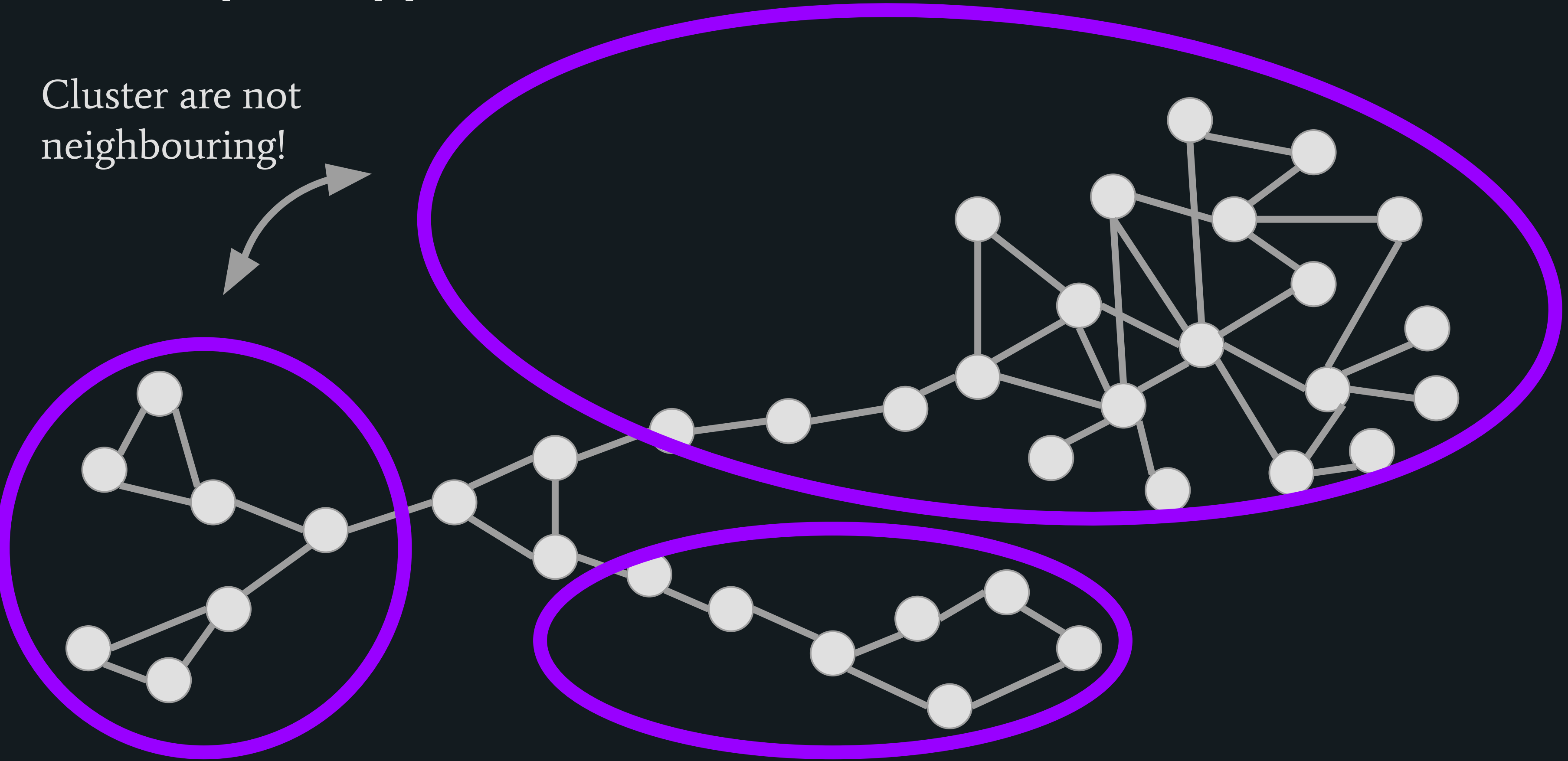


Principled approach

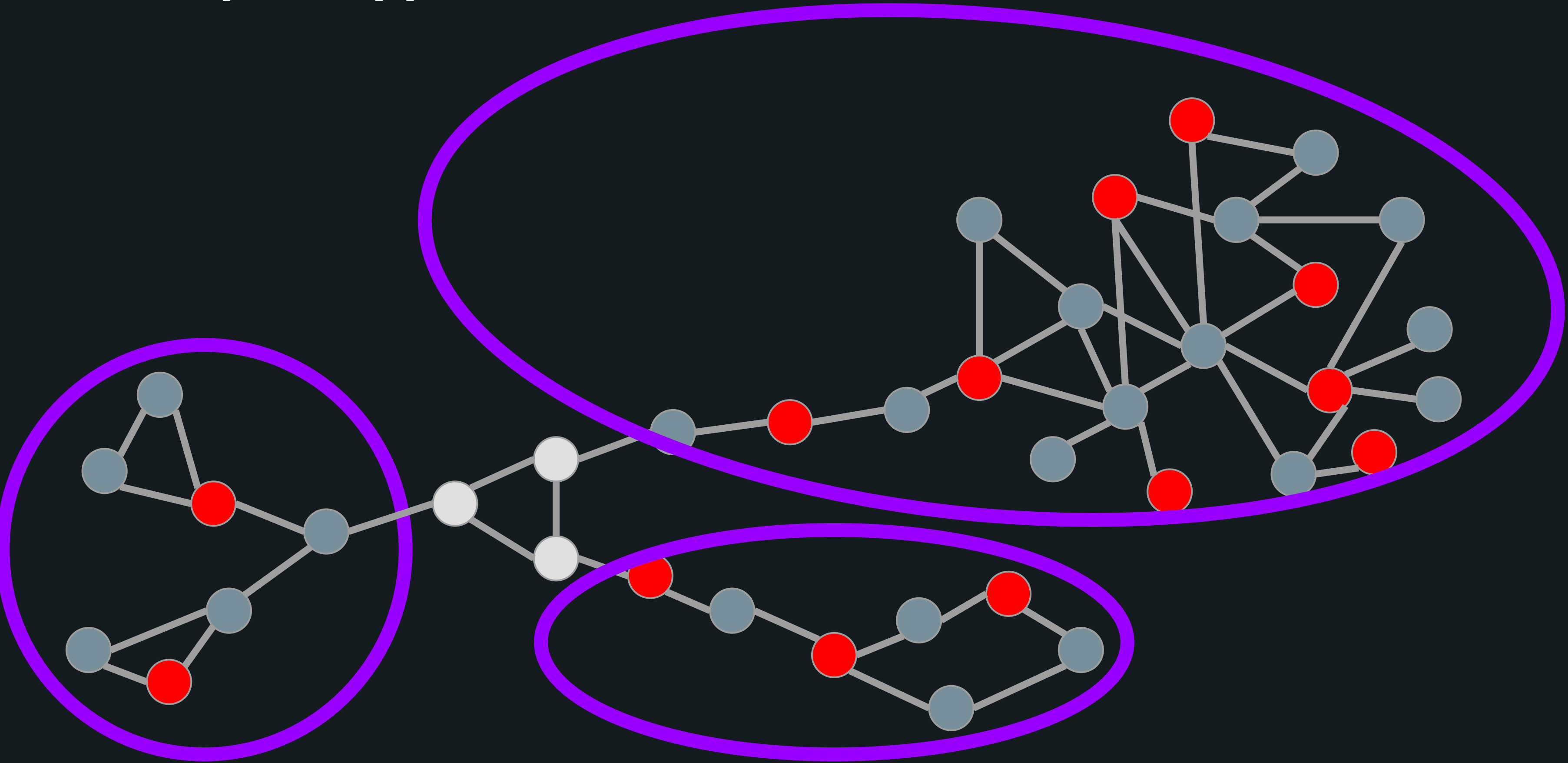


Principled approach

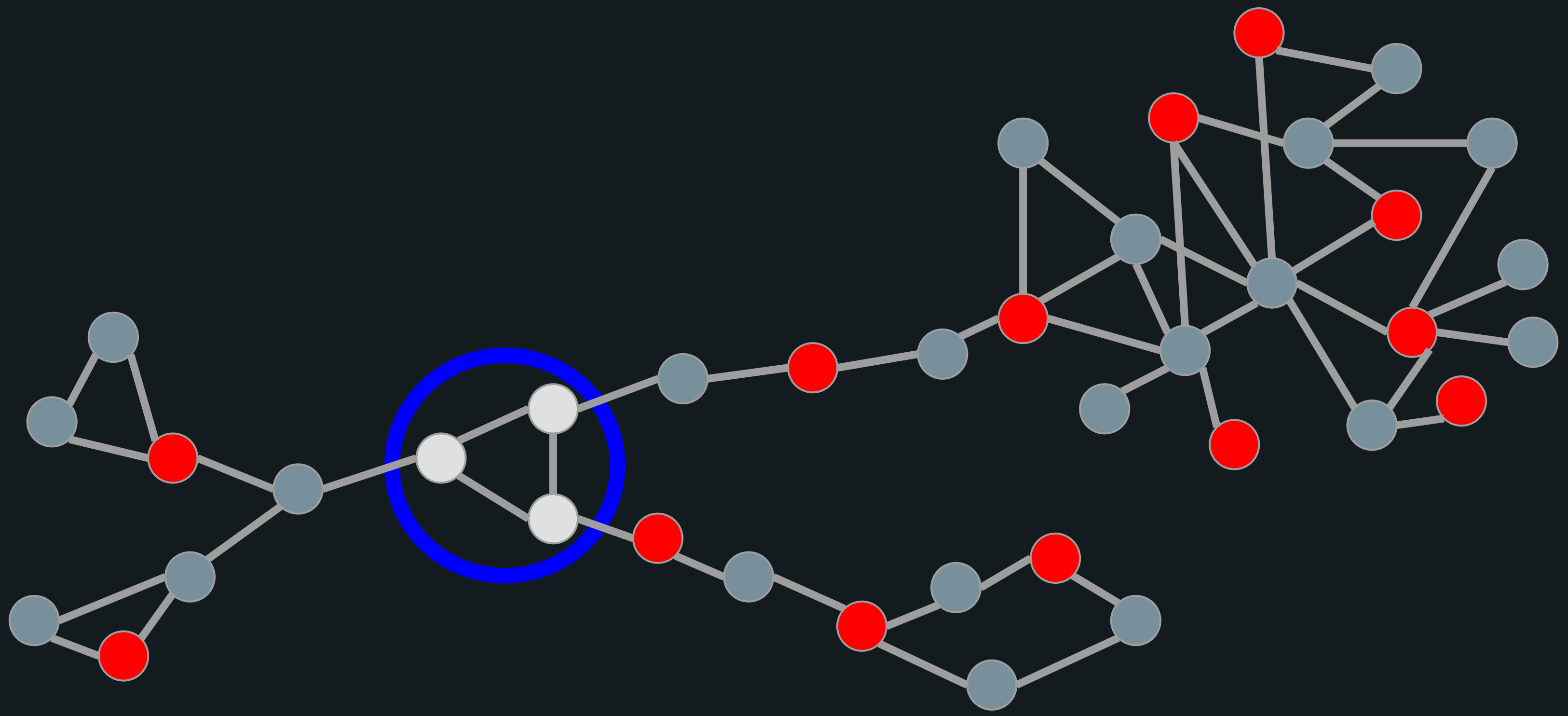
Cluster are not neighbouring!



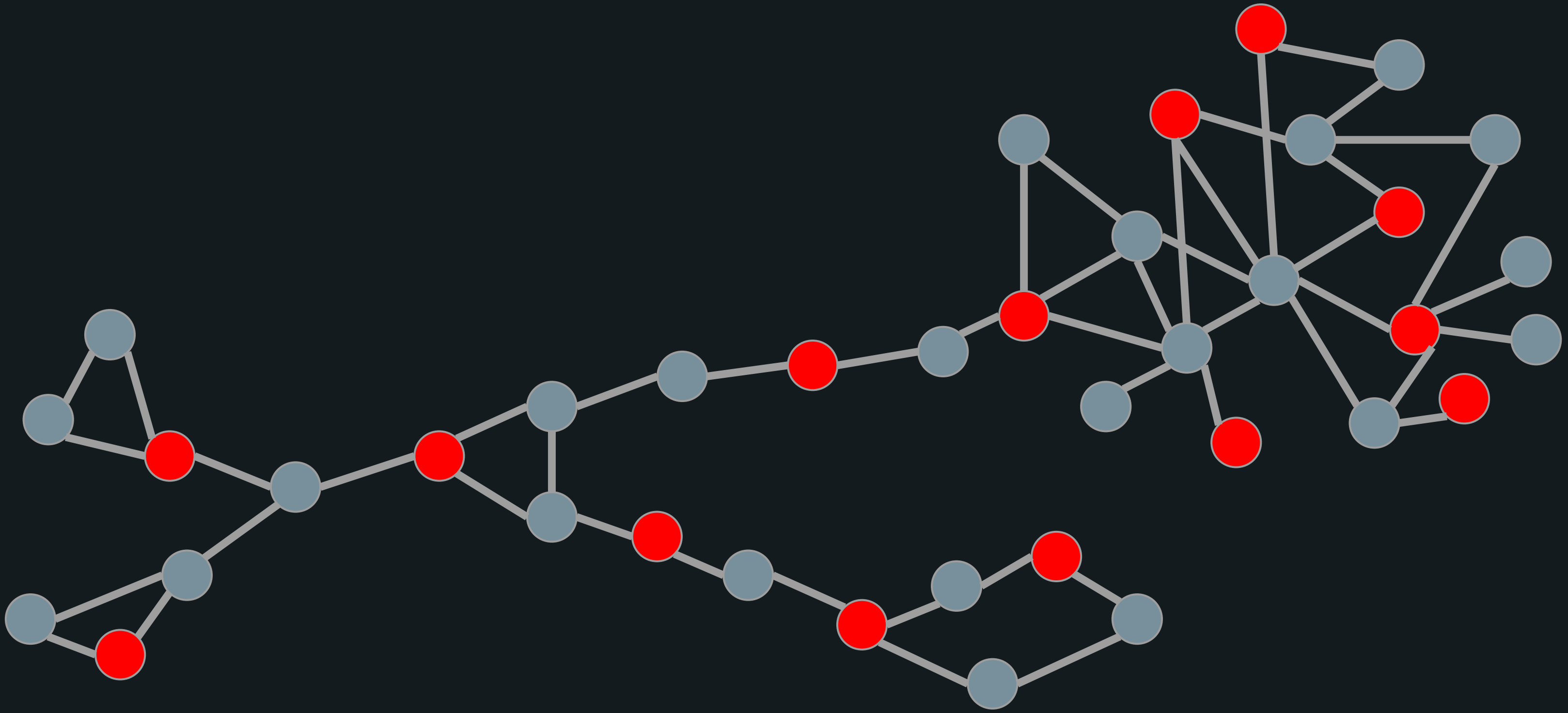
Principled approach



Principled approach



Principled approach



We need to…

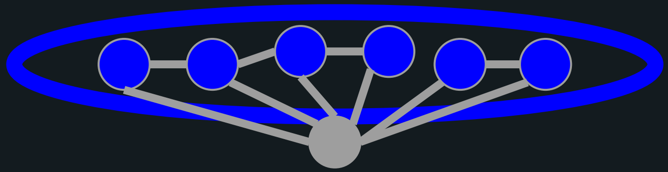
…partition the underlying graph into non-neighbouring poly(log n)-diameter clusters that cover at least half of the vertices.

Then we just solve inside clusters and iterate this O(log(n)) times.

Network decomposition with **C** colors and diameter **D**:

Coloring of the vertices with **C** colors, such that each component induced by a particular color has diameter at most **D**

We need to...



...partition the underlying graph into poly(log n)-diameter clusters that cover at least half of the vertices.

Then we just solve inside clusters and iterate this O(log(n)) times.

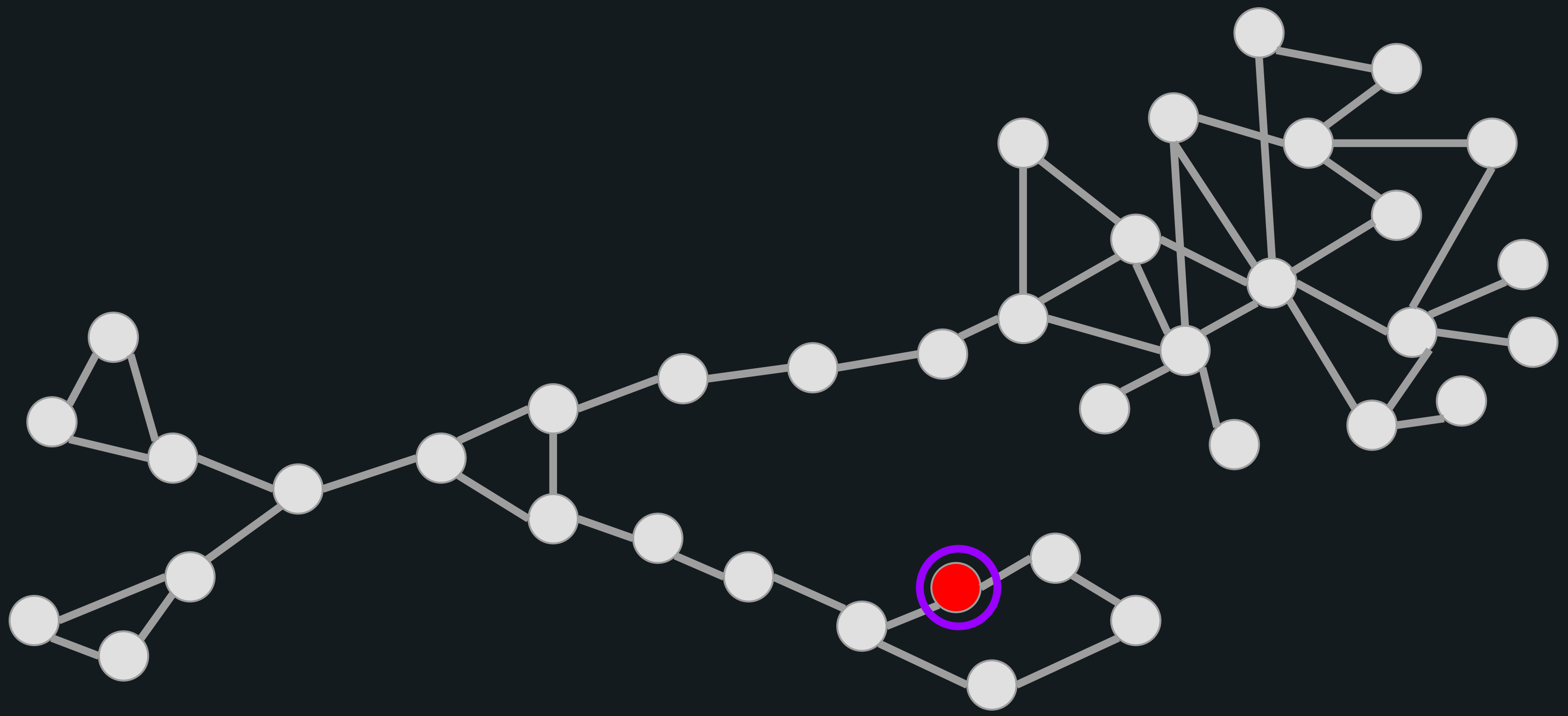
Network decomposition with C colors and diameter D:

Coloring of the vertices with C colors, such that each component induced by a particular color ~~has diameter at most D~~

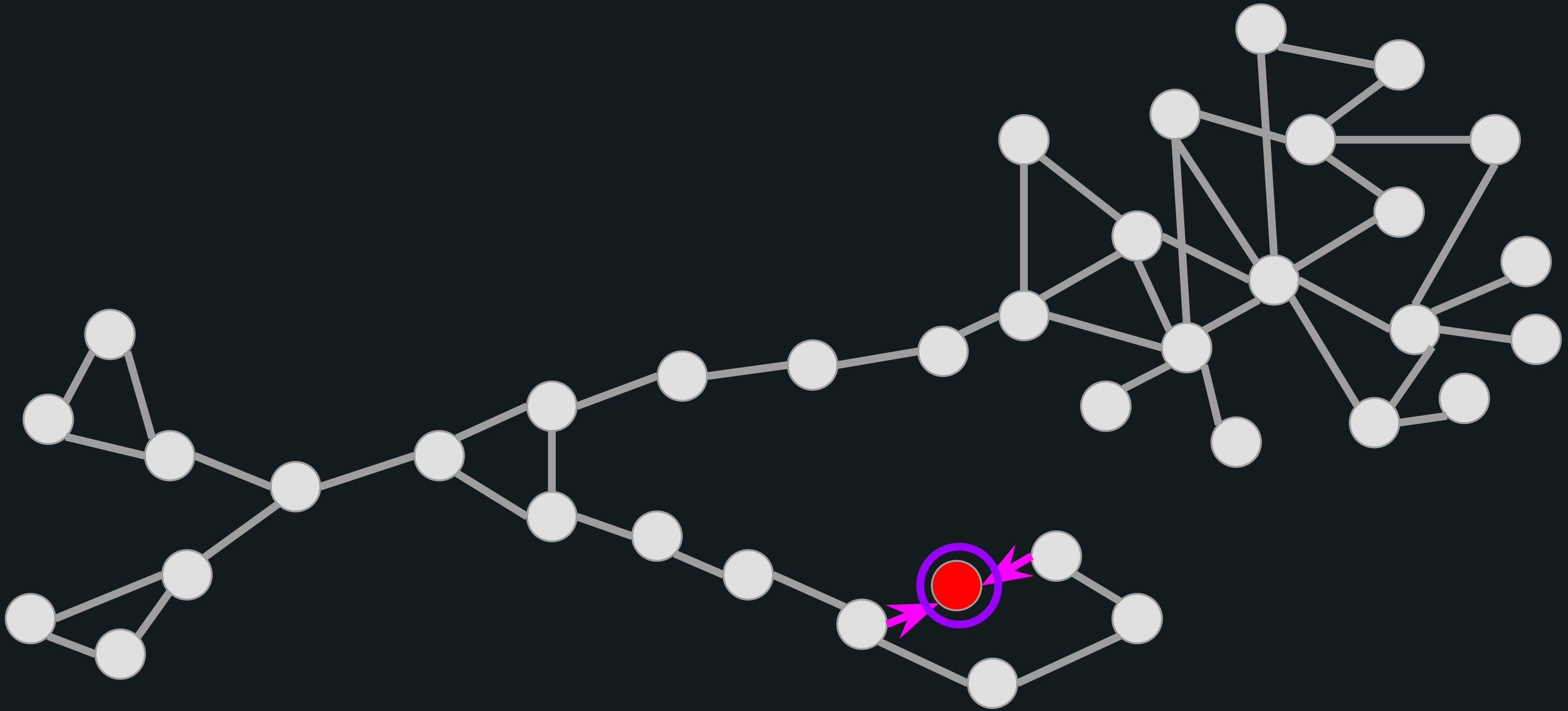
...satisfies that any two of its vertices are at most D hops apart in the original graph.

Sequential algorithm

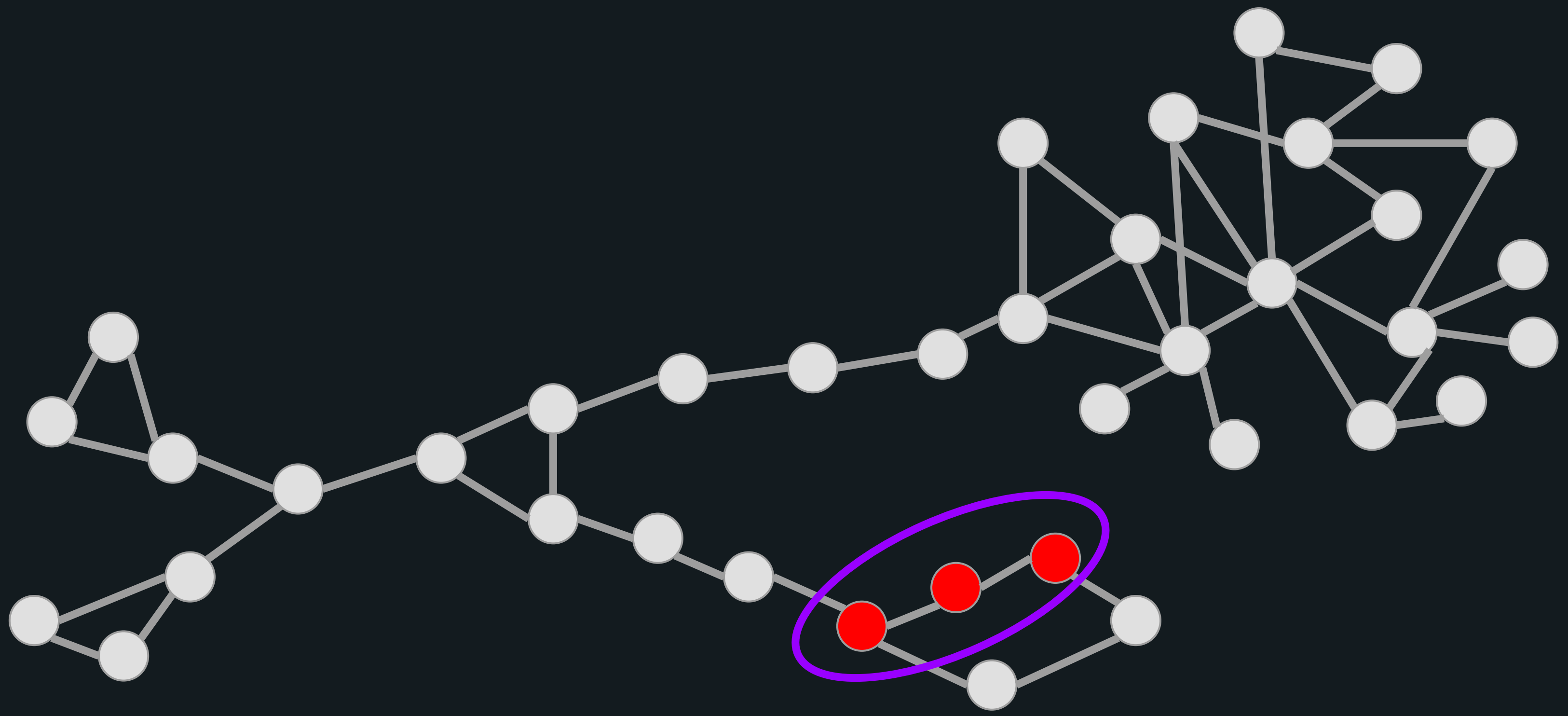
Sequential algorithm



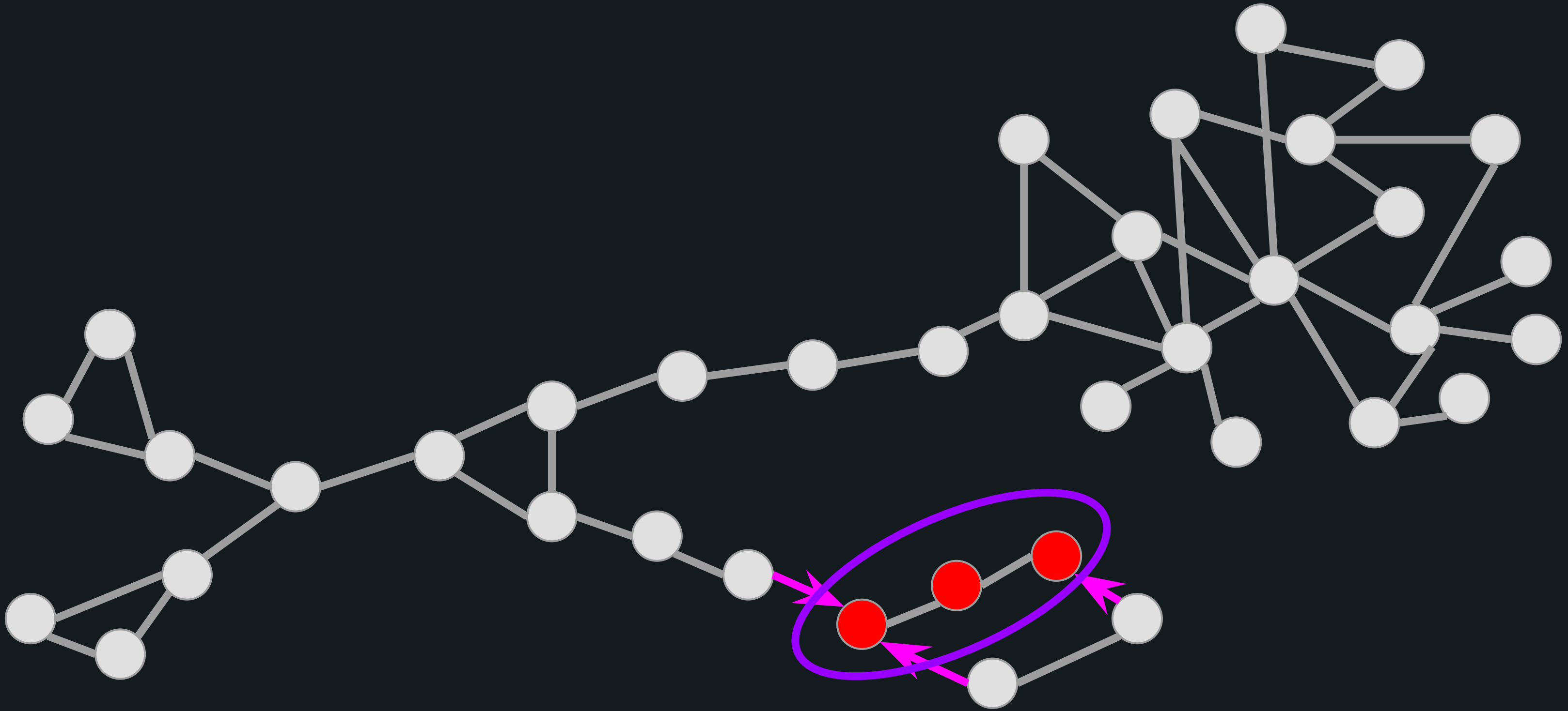
Sequential algorithm



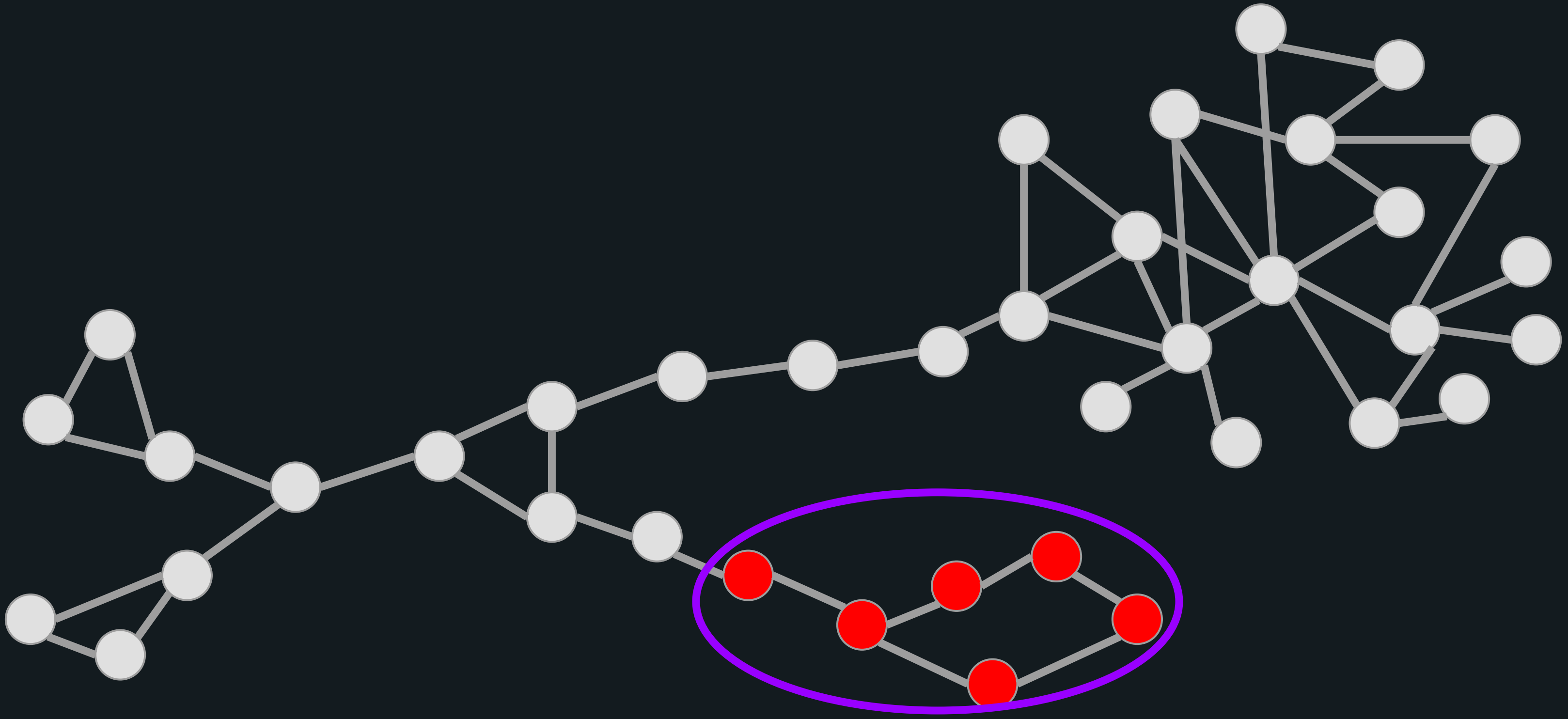
Sequential algorithm



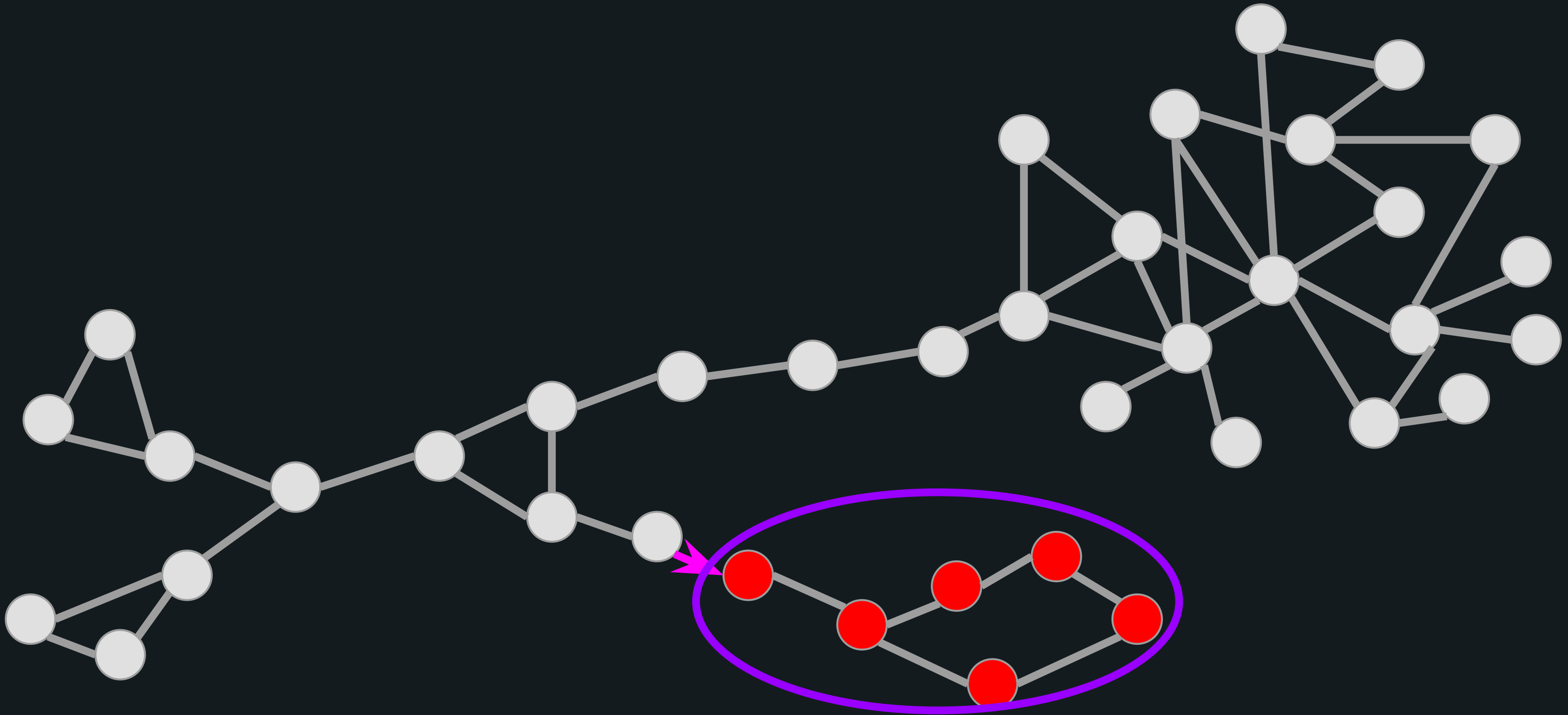
Sequential algorithm



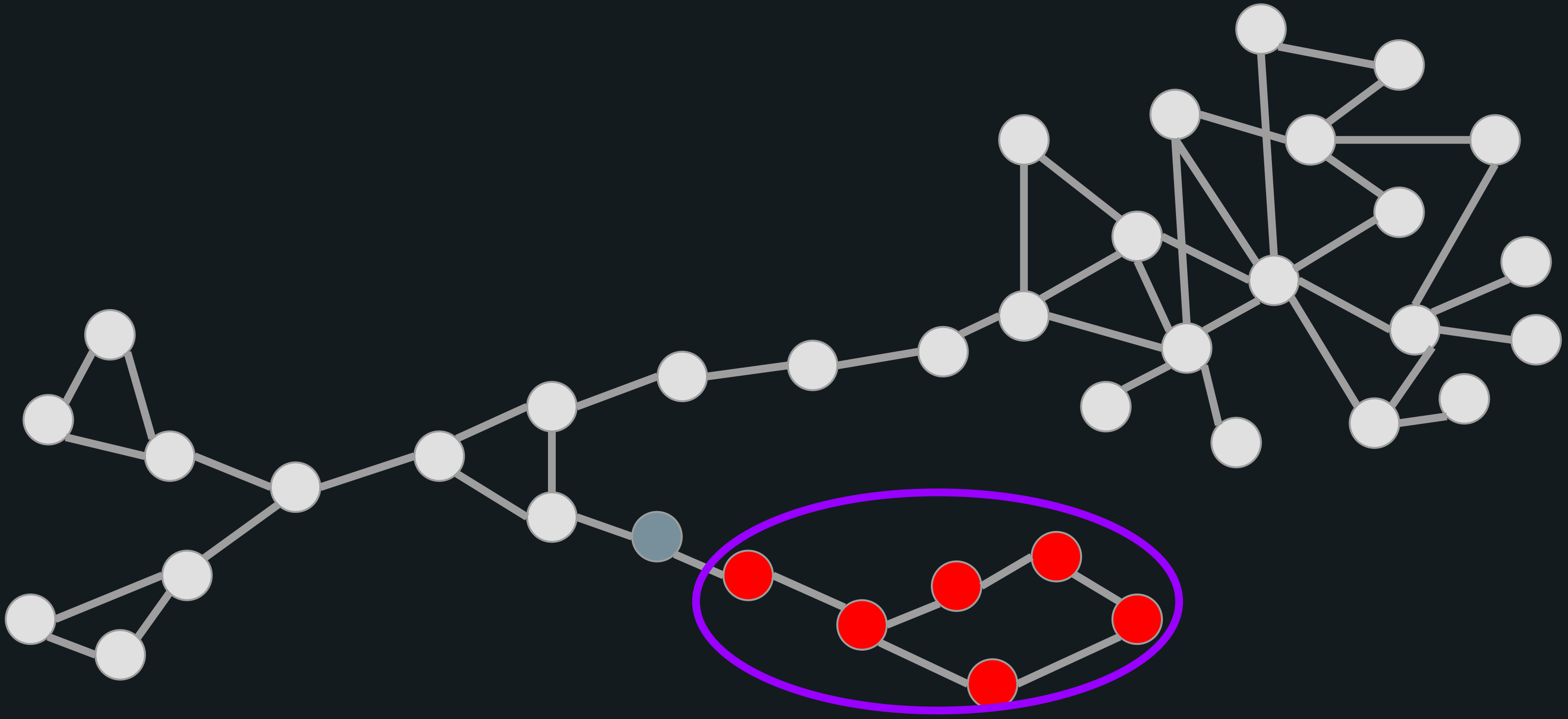
Sequential algorithm



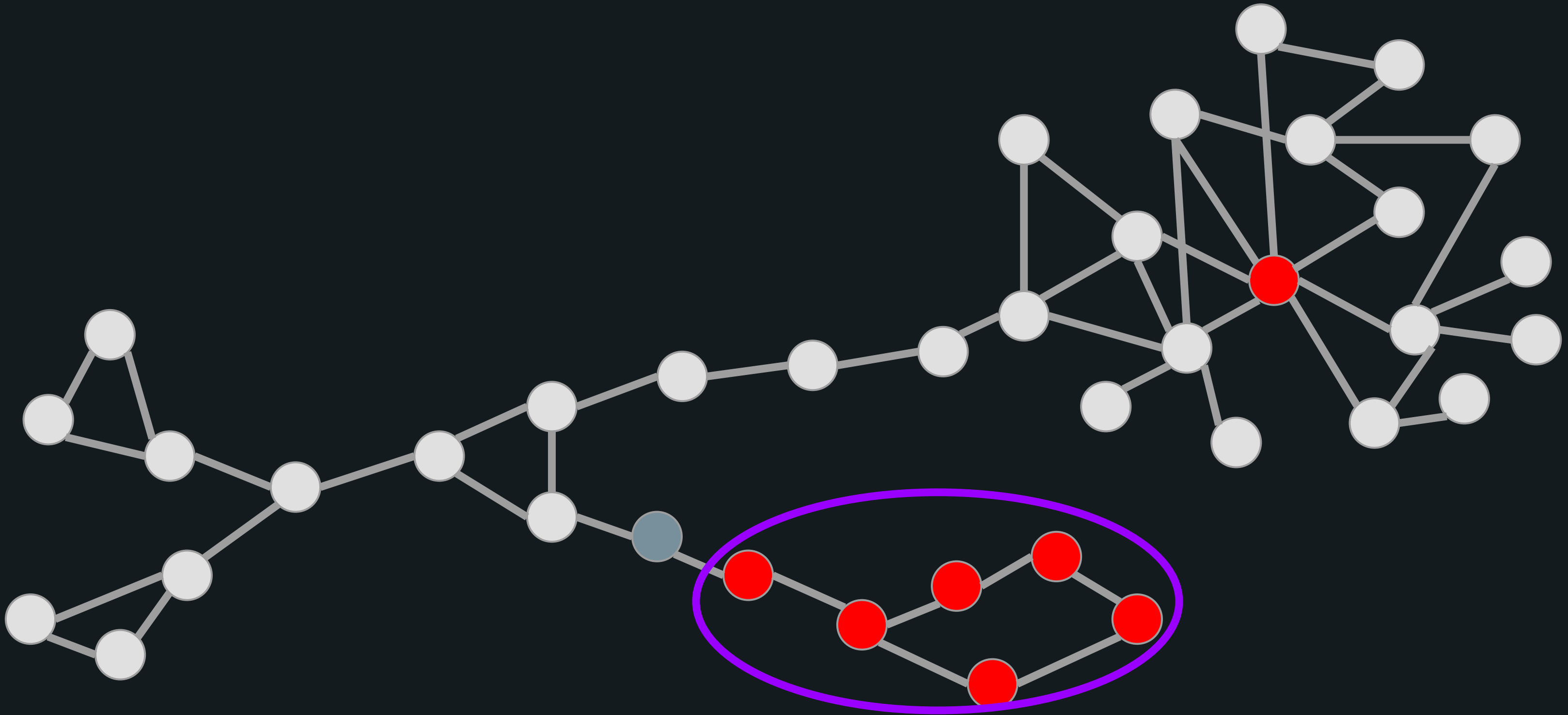
Sequential algorithm



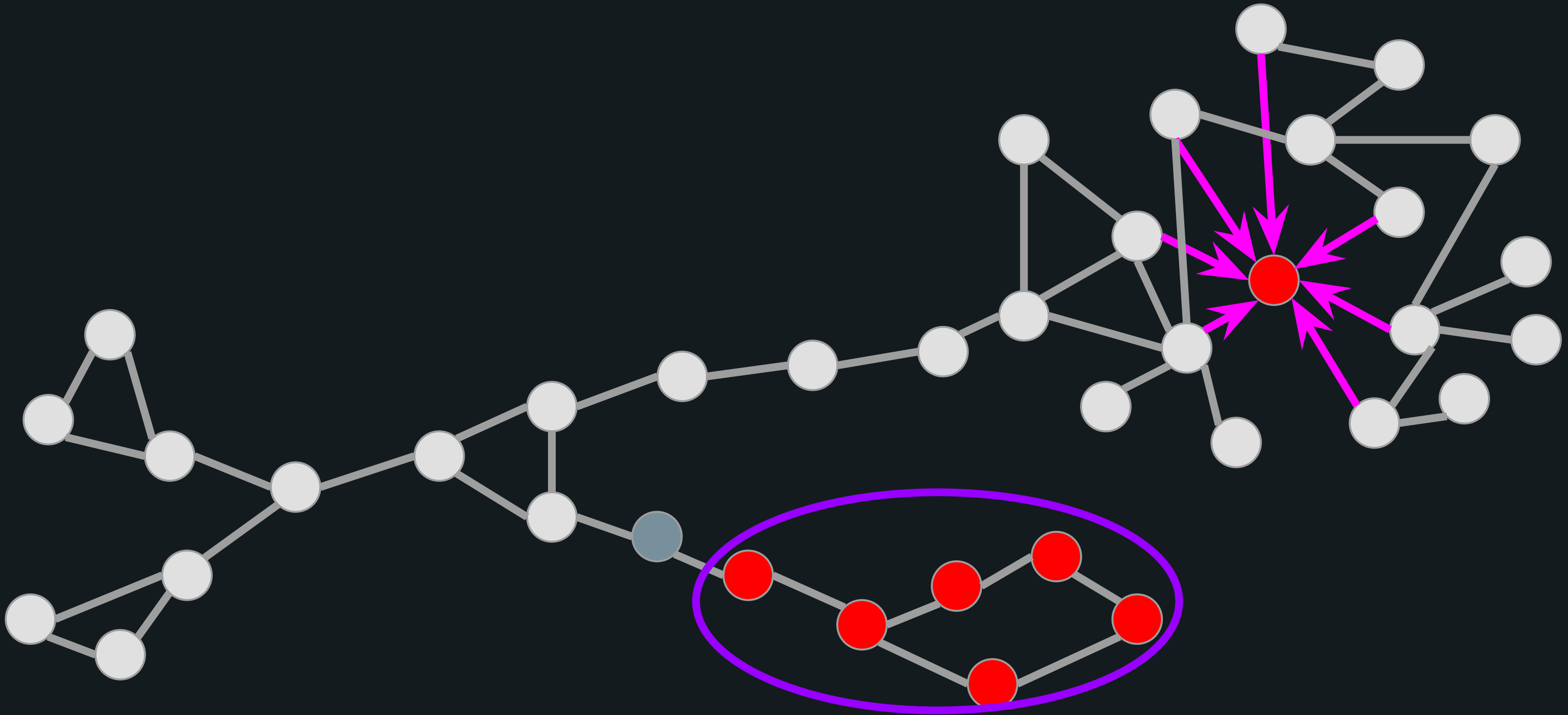
Sequential algorithm



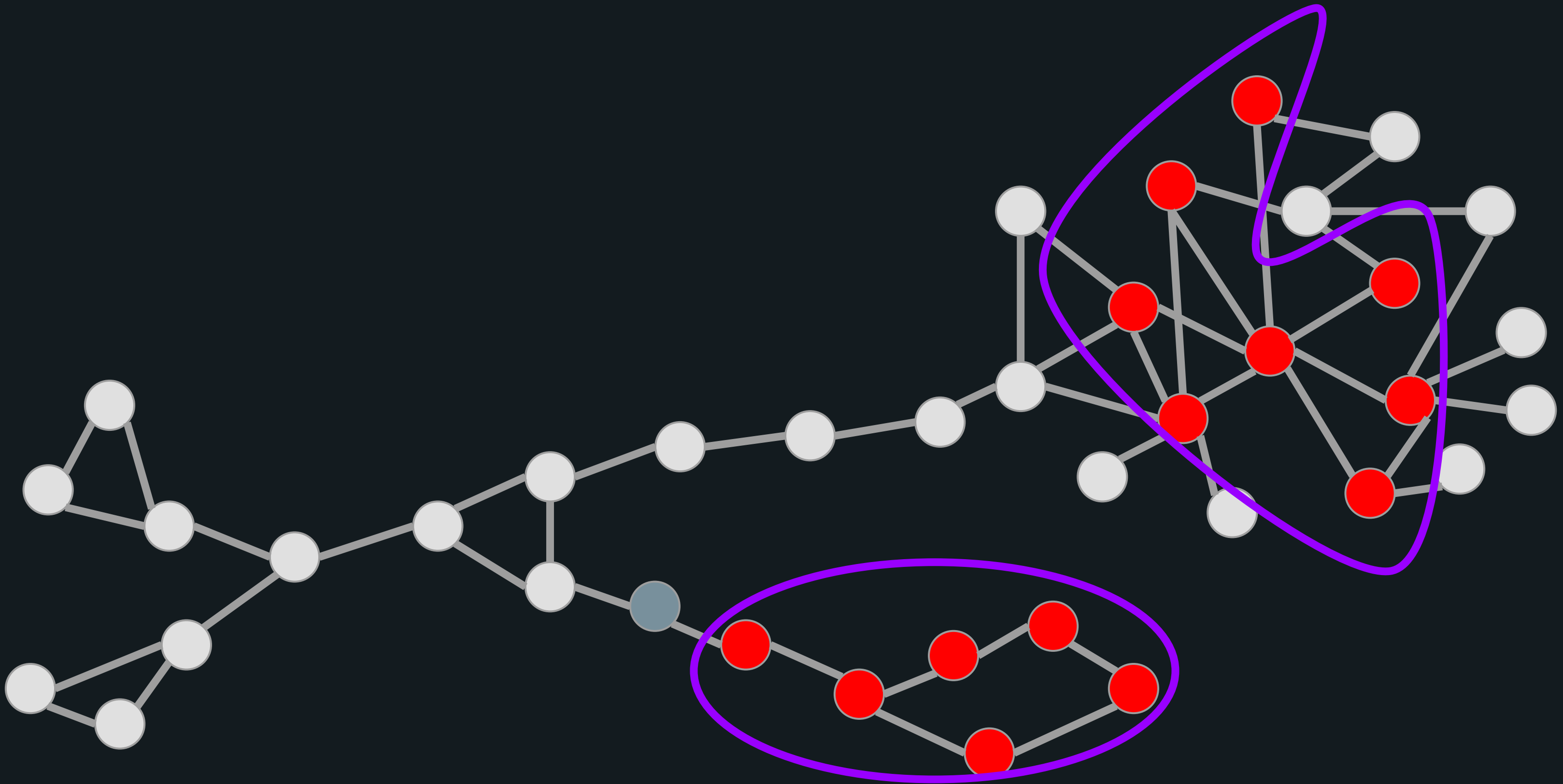
Sequential algorithm



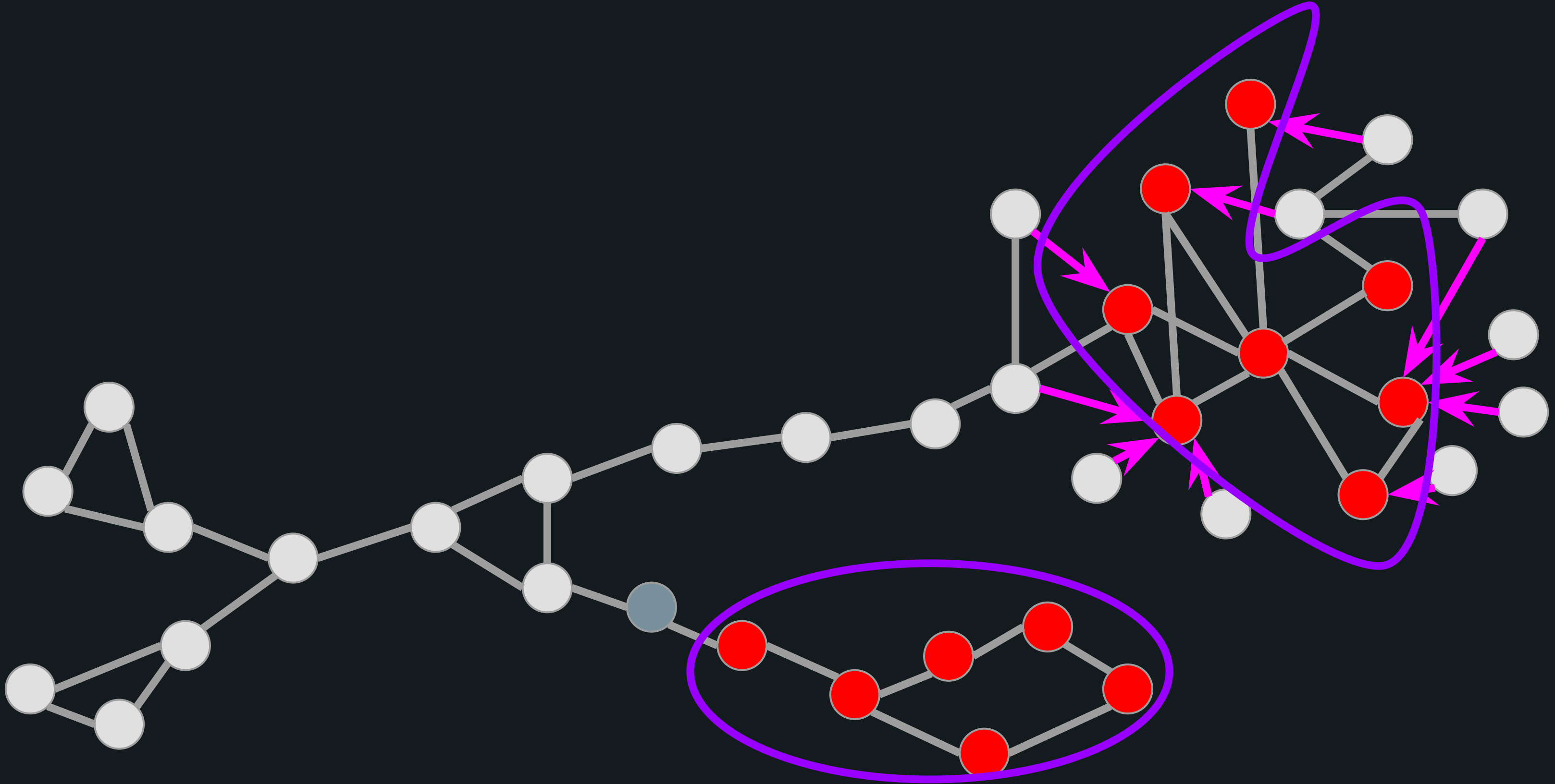
Sequential algorithm



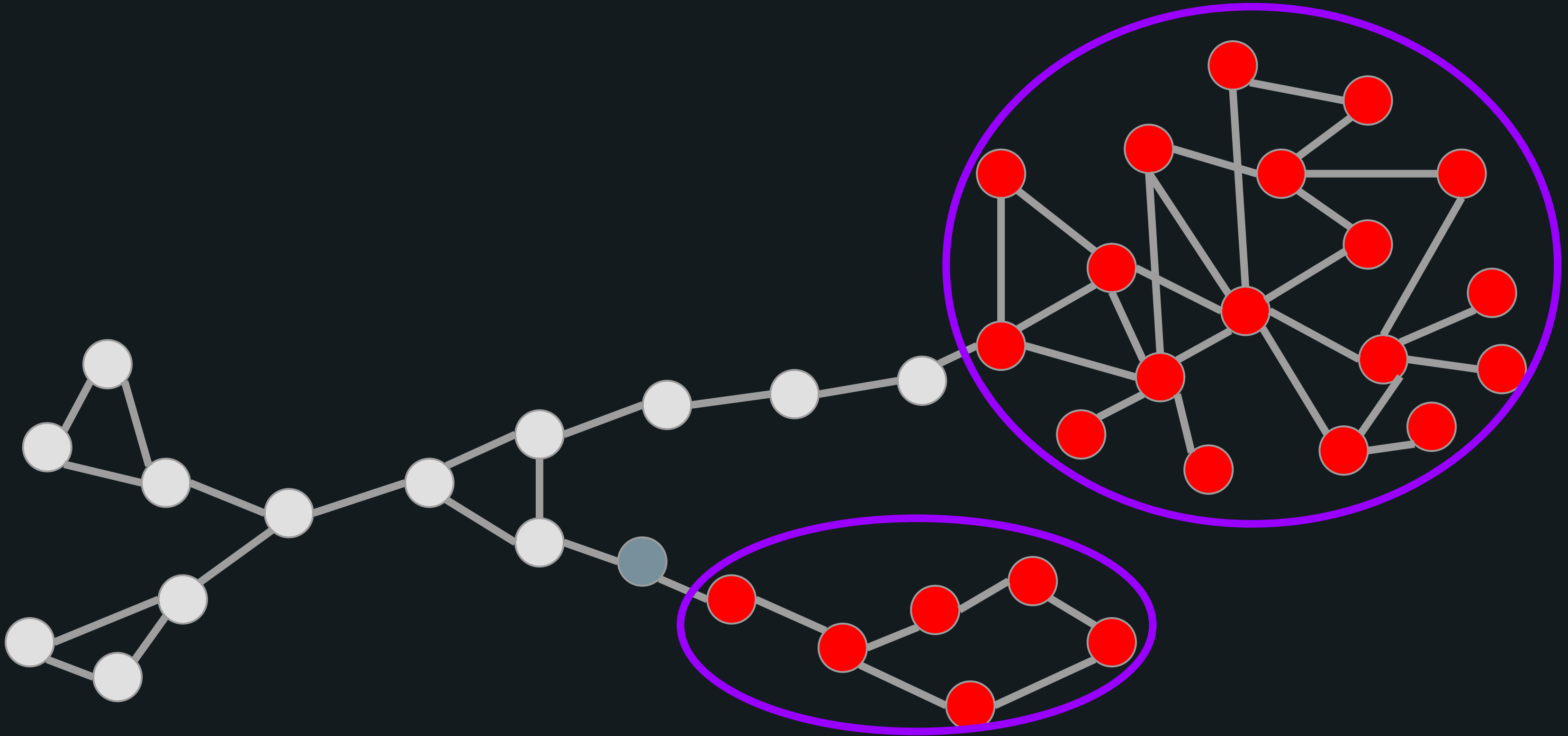
Sequential algorithm



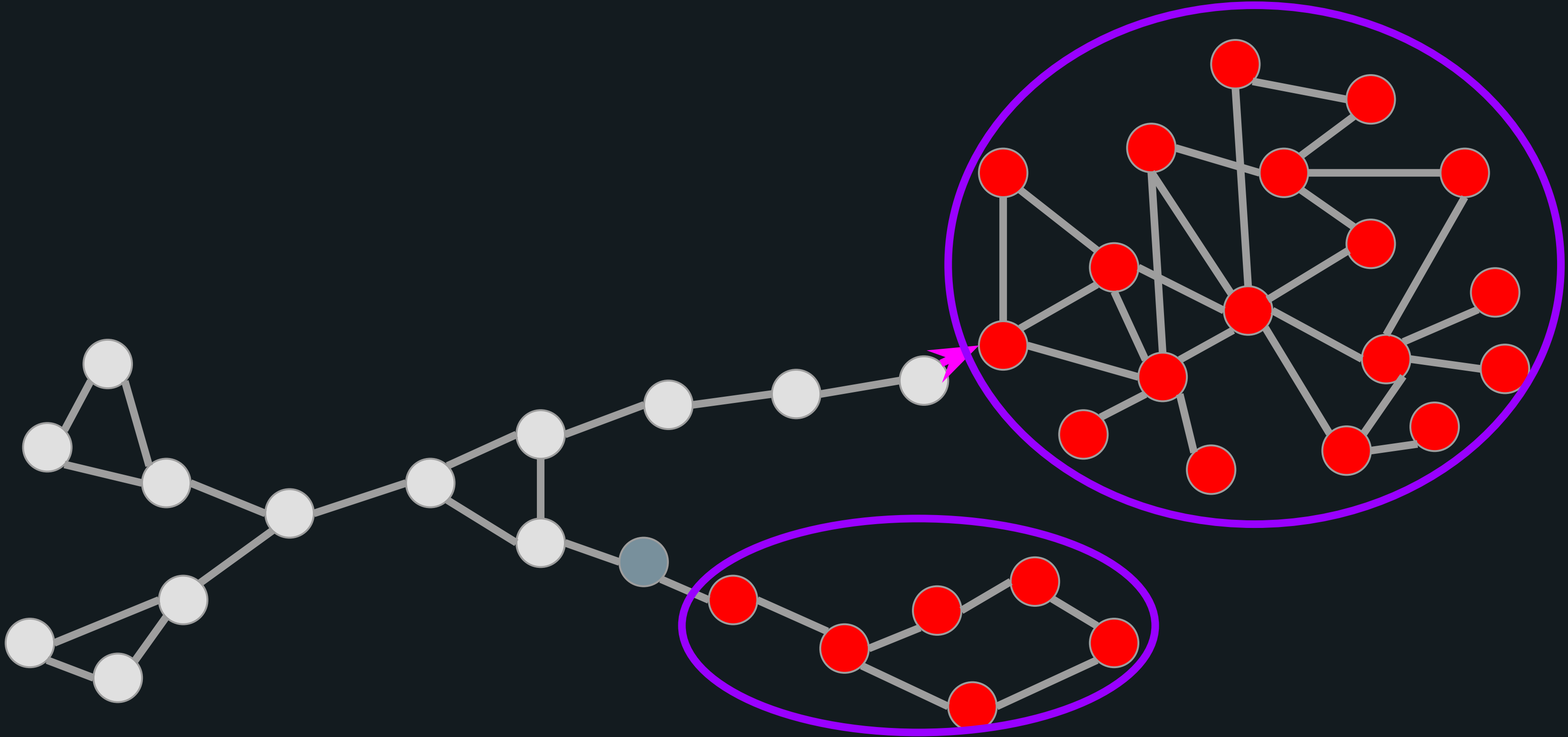
Sequential algorithm



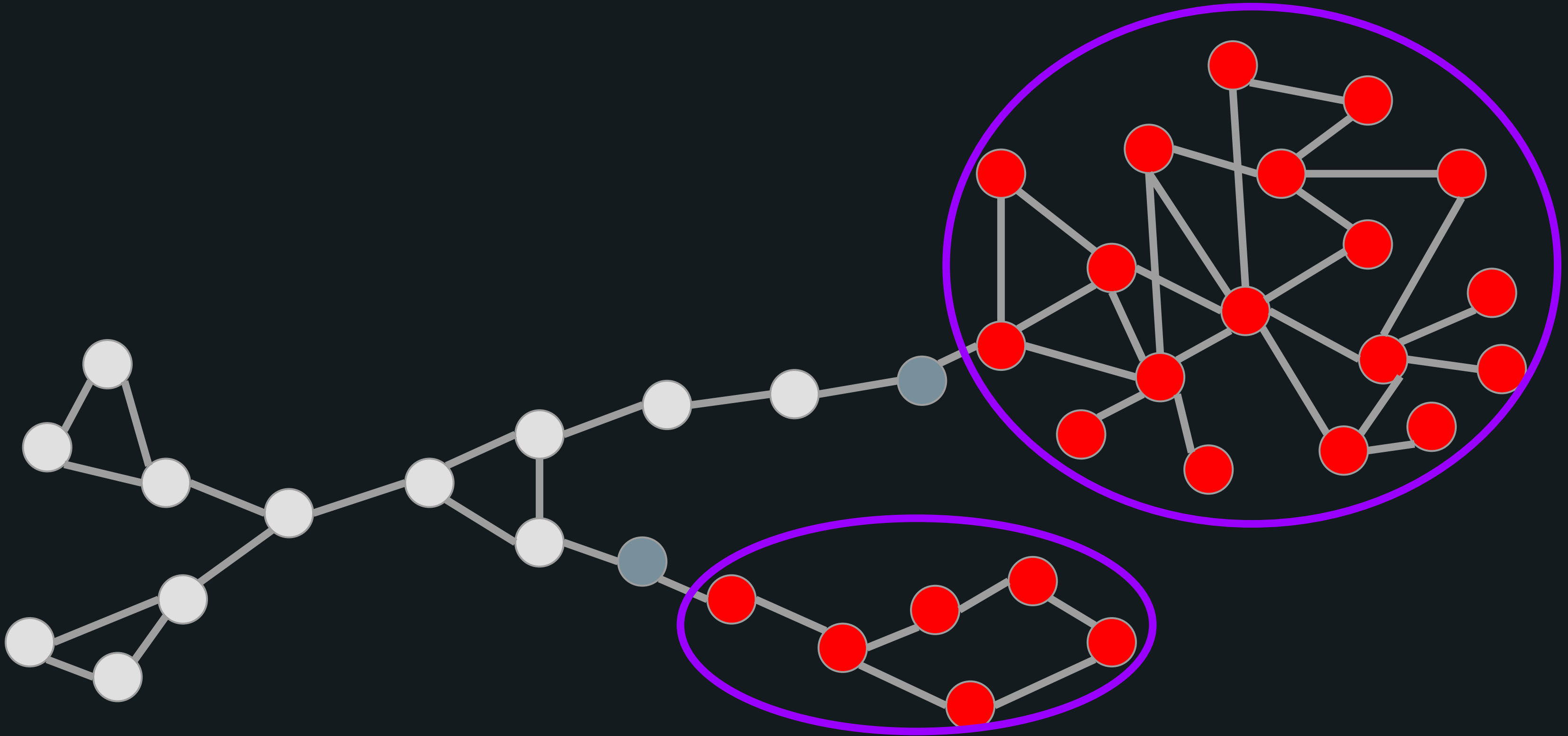
Sequential algorithm



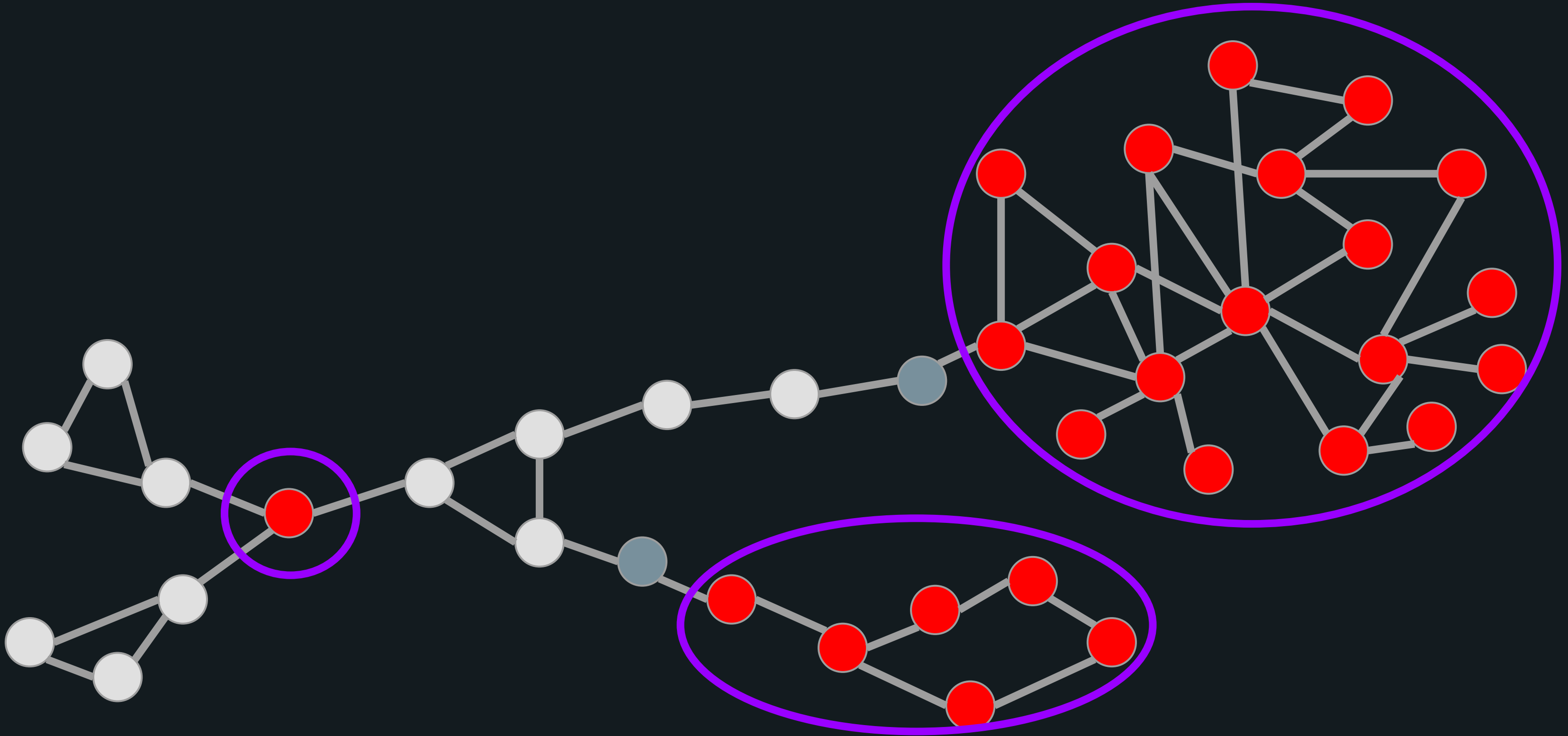
Sequential algorithm



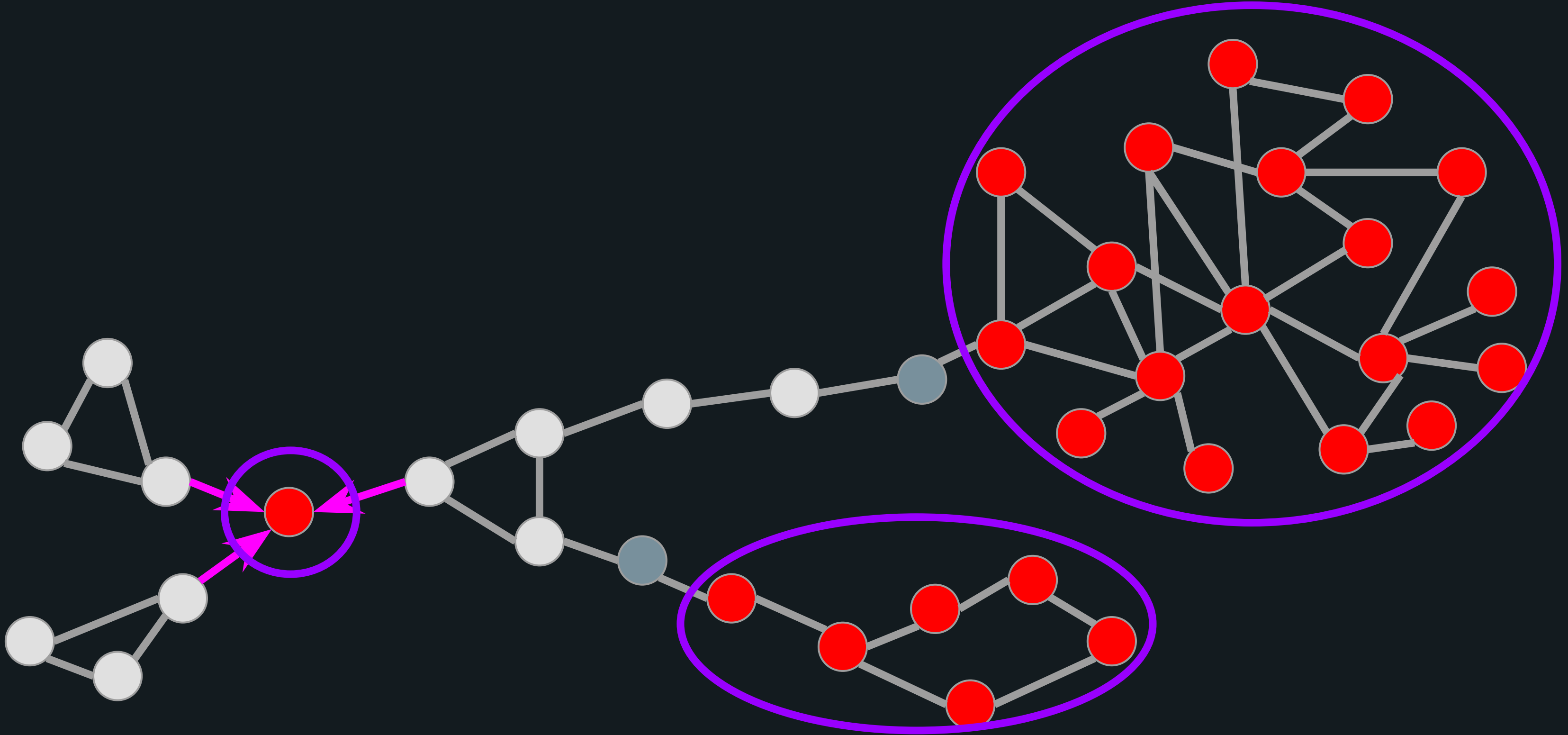
Sequential algorithm



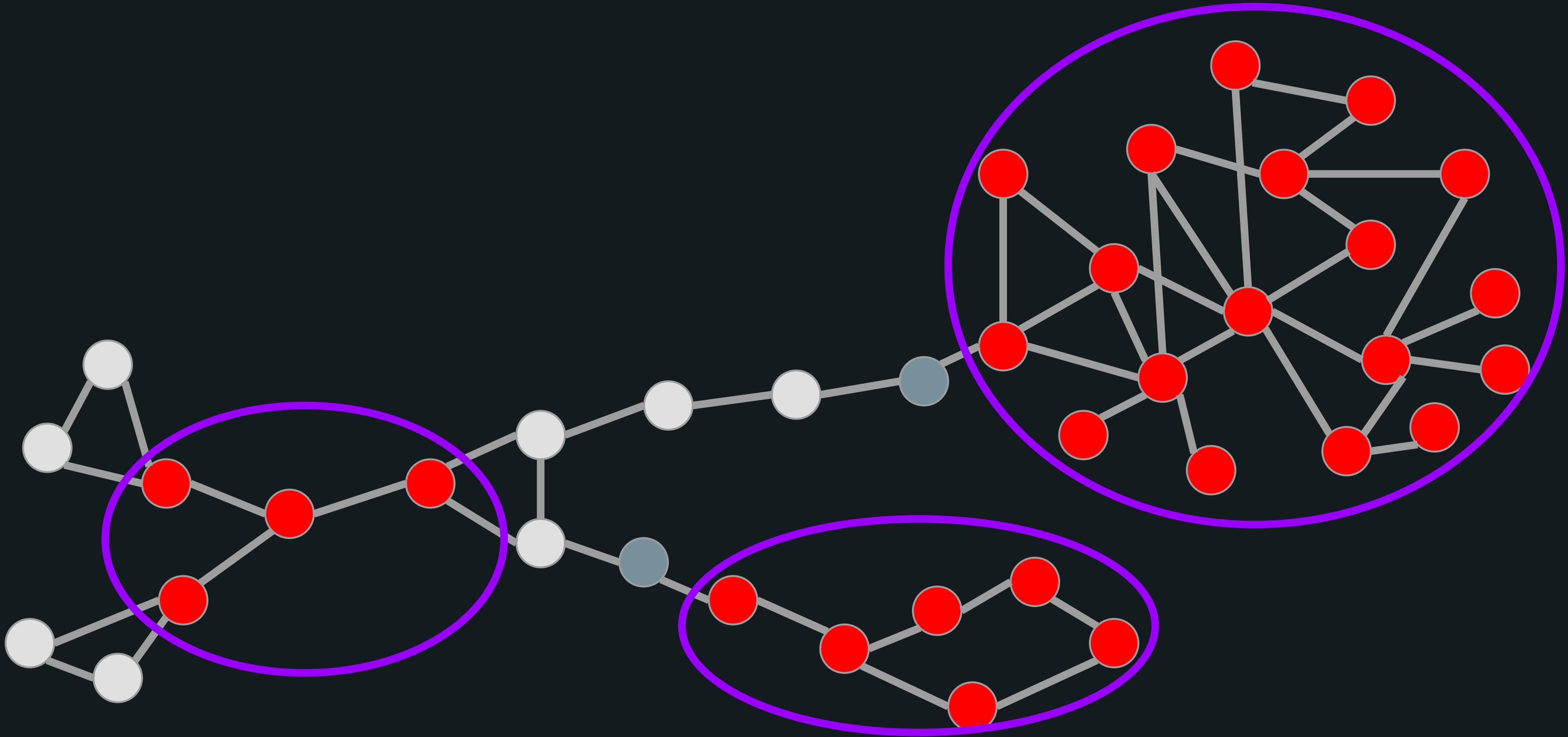
Sequential algorithm



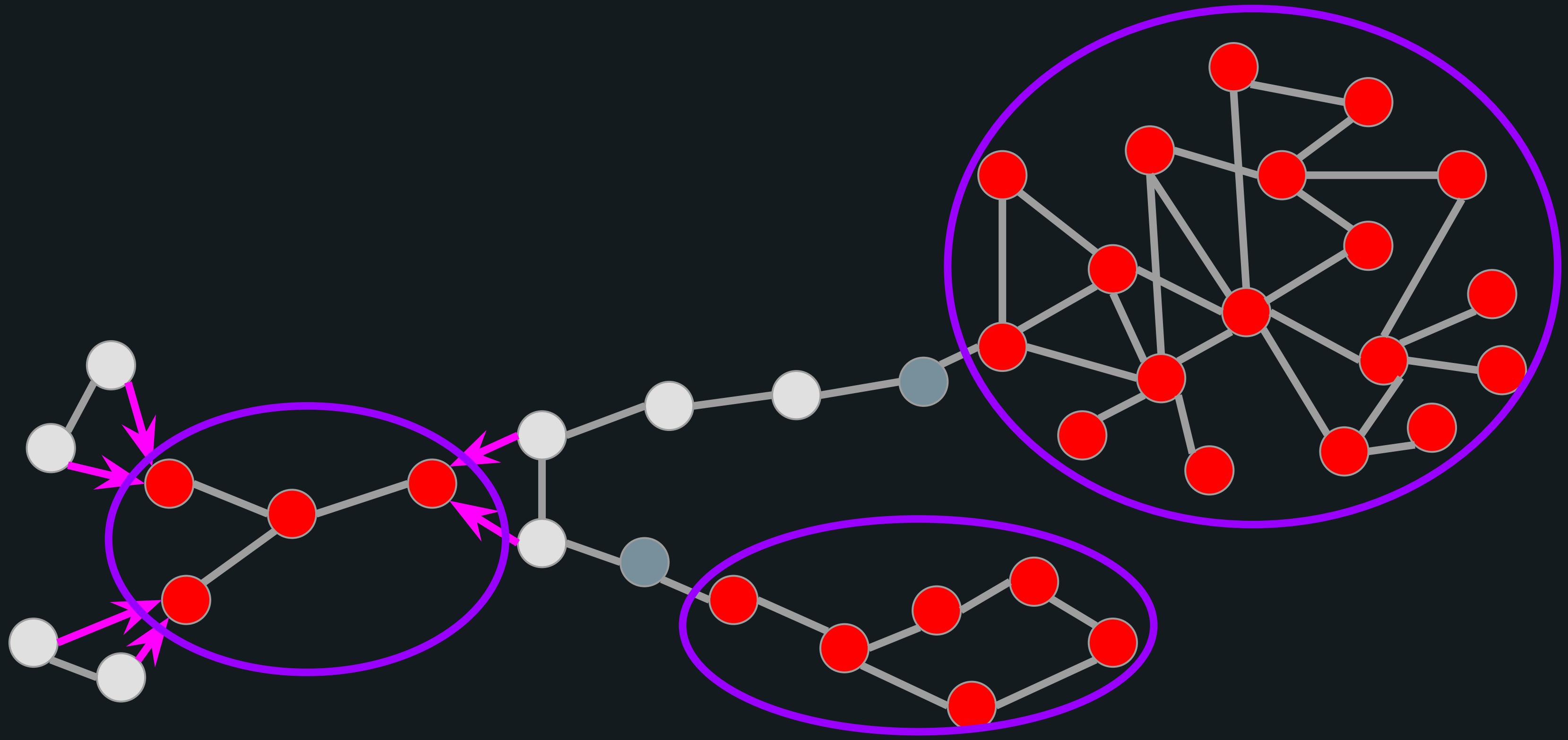
Sequential algorithm



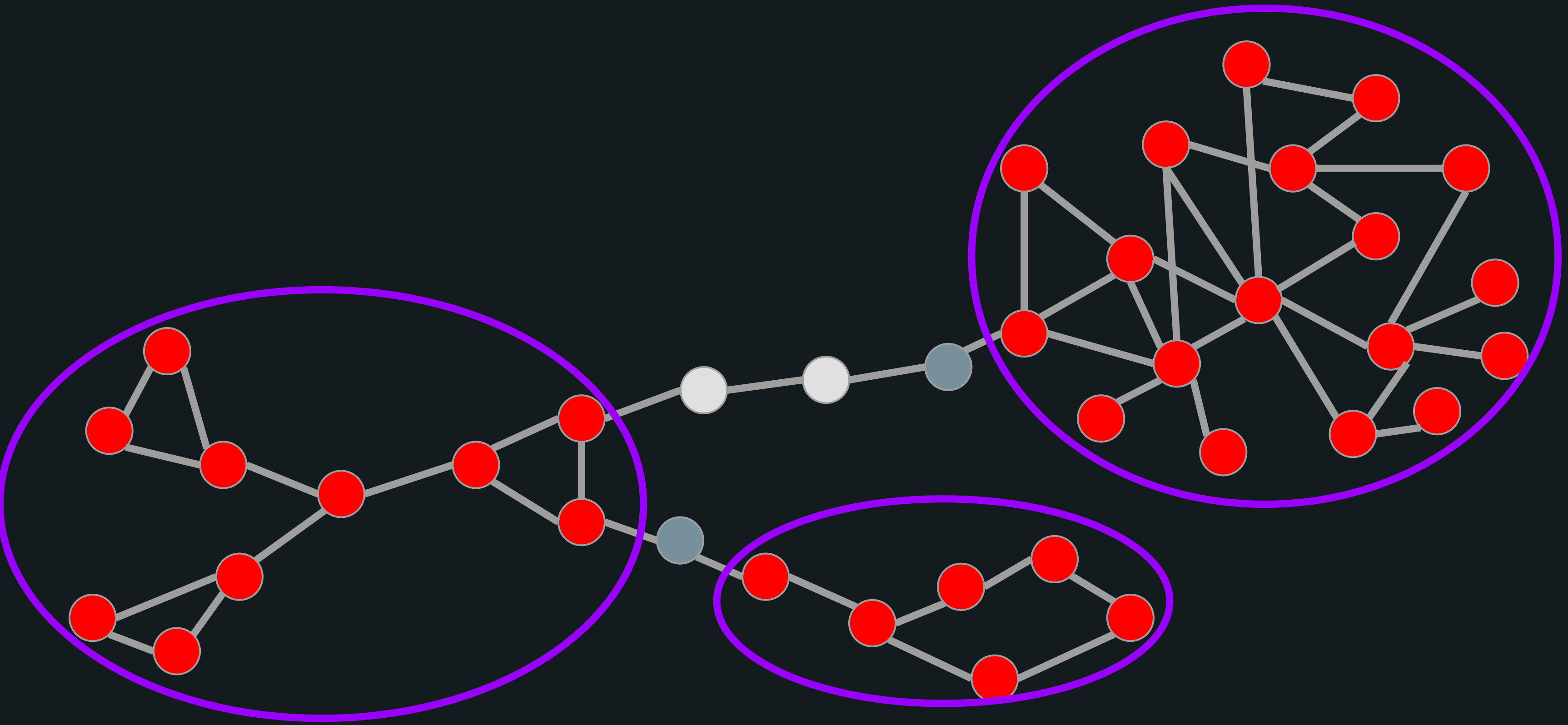
Sequential algorithm



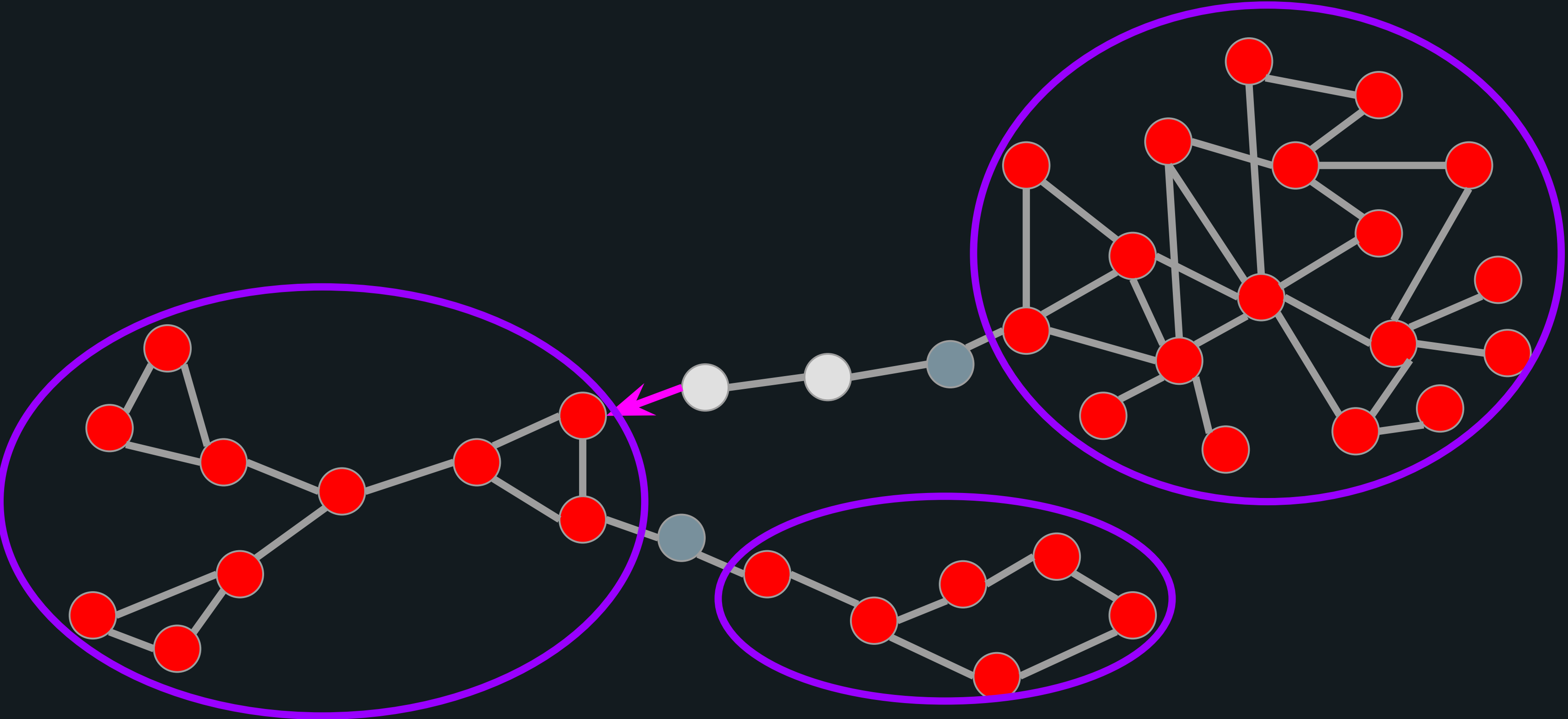
Sequential algorithm



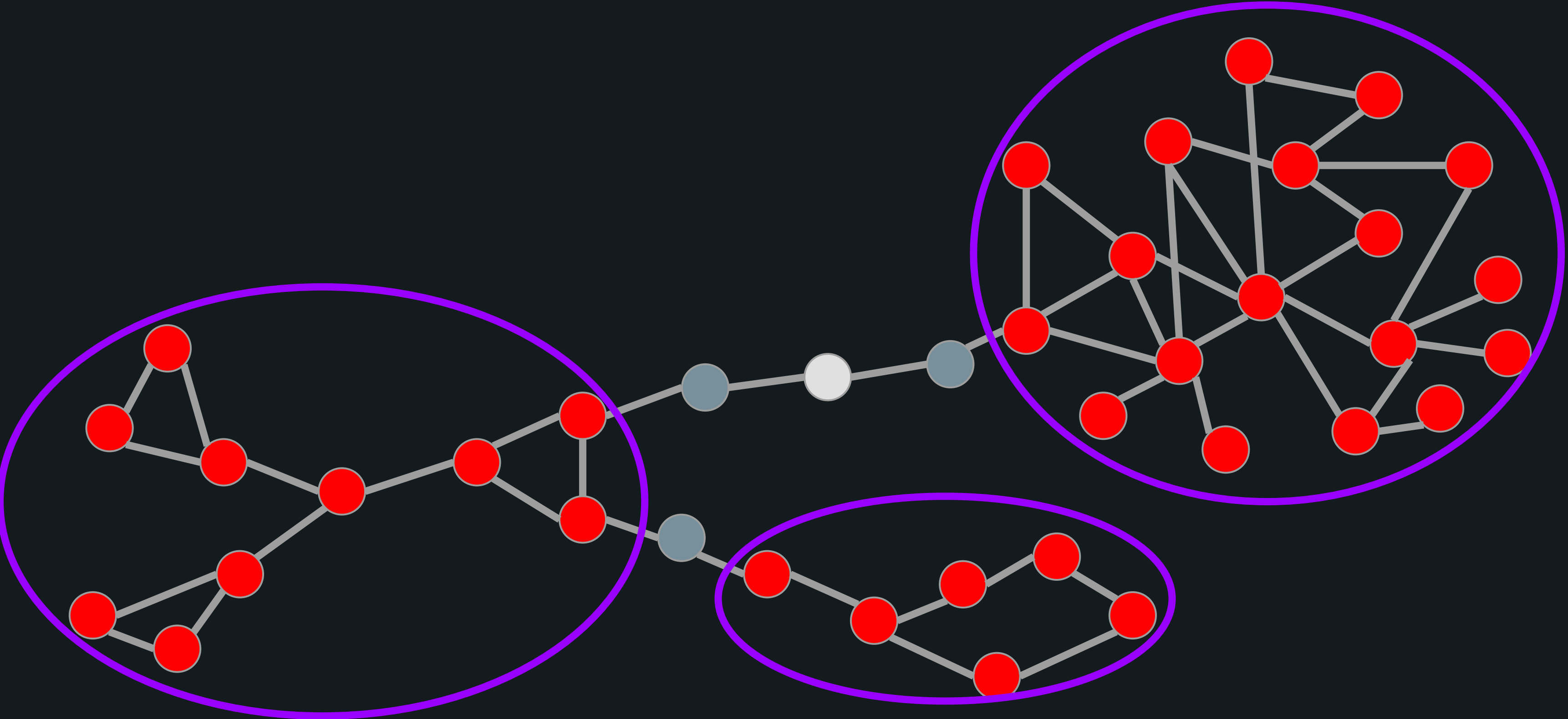
Sequential algorithm



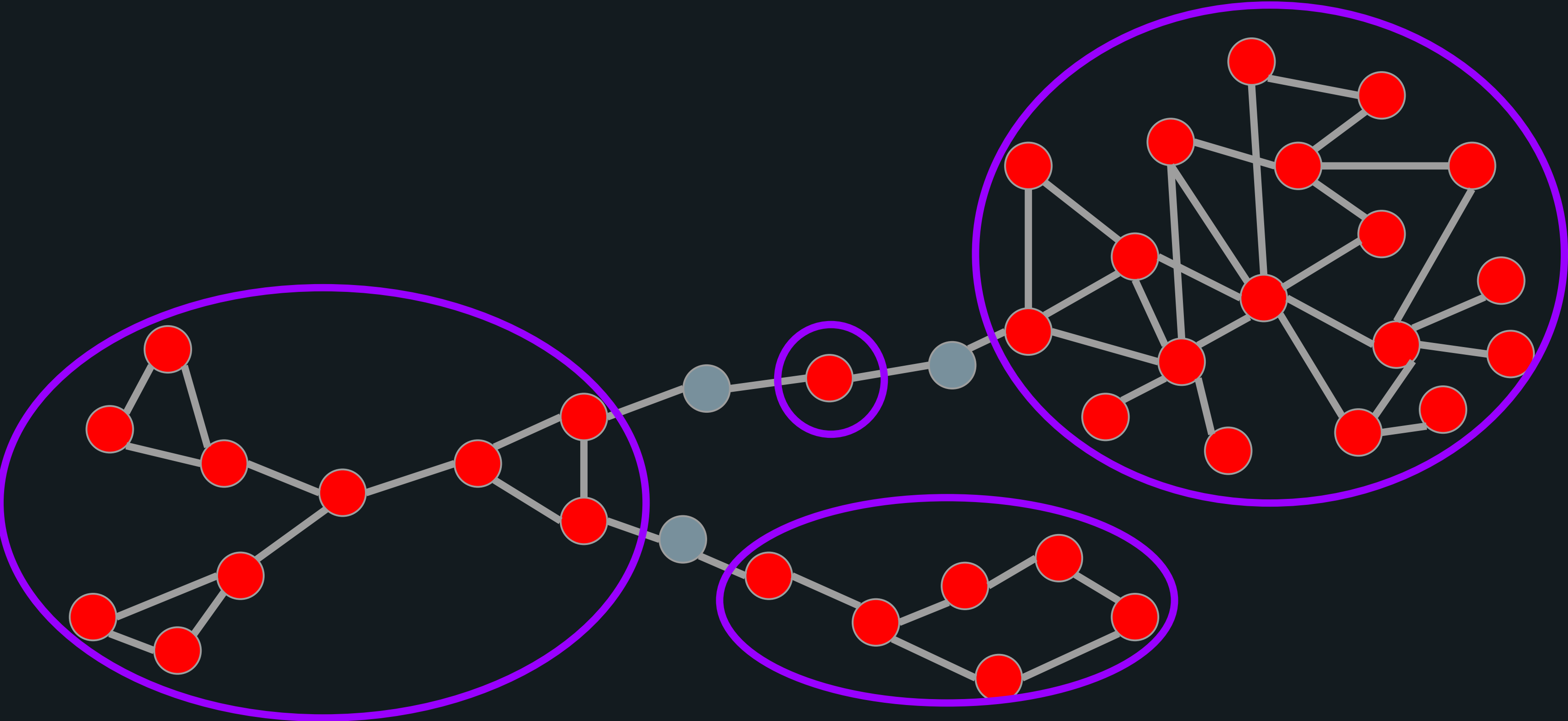
Sequential algorithm



Sequential algorithm

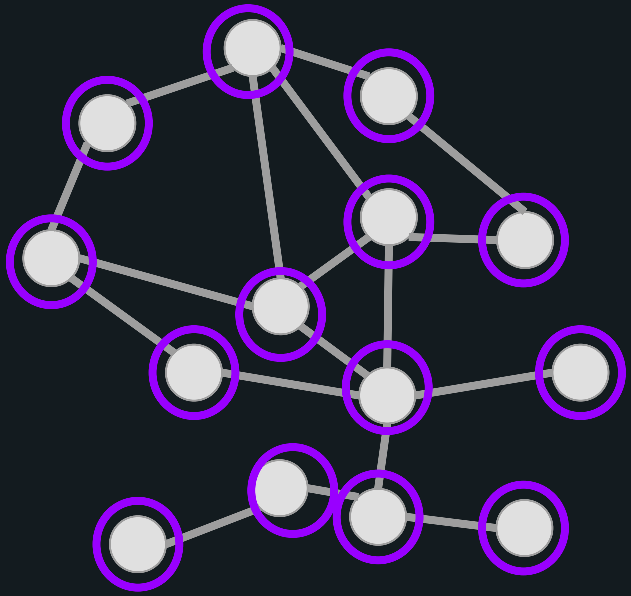


Sequential algorithm



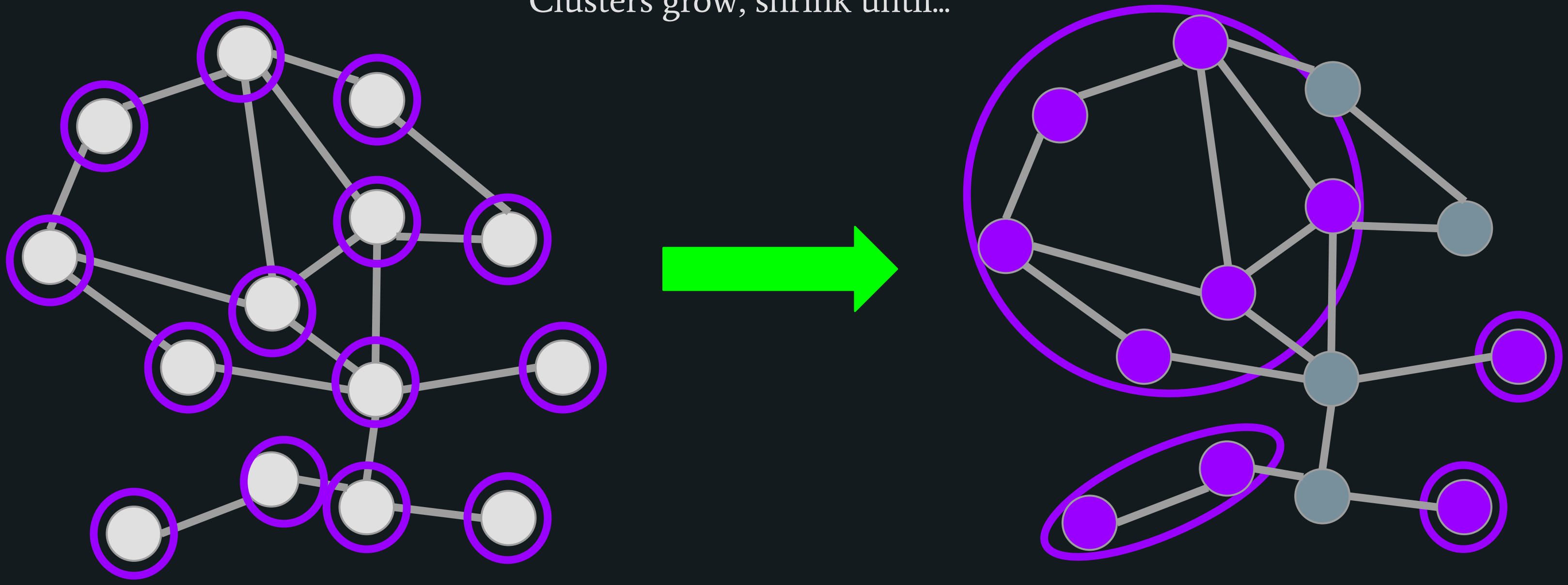
Network decomposition:

Network decomposition:

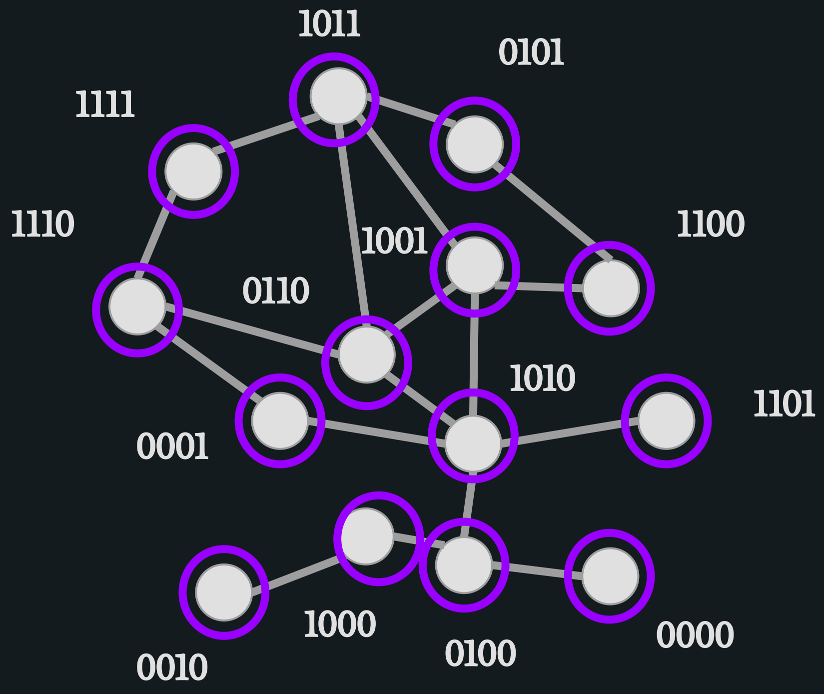


Network decomposition:

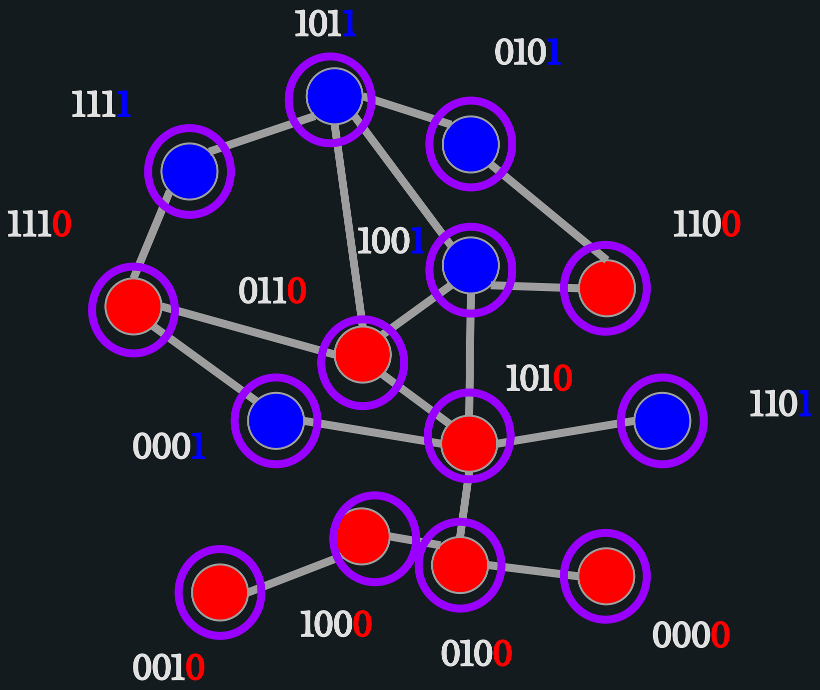
Clusters grow, shrink until...



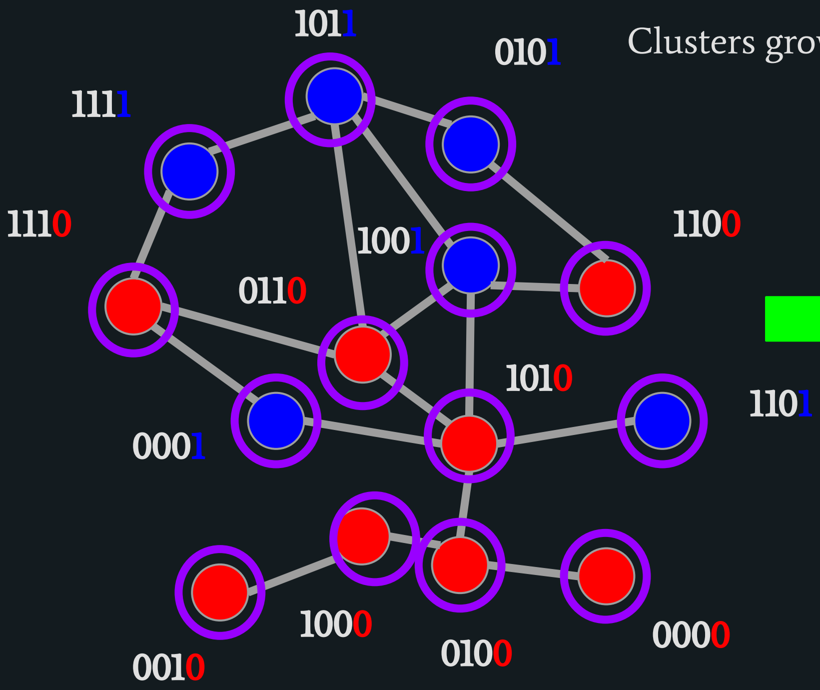
Network decomposition:



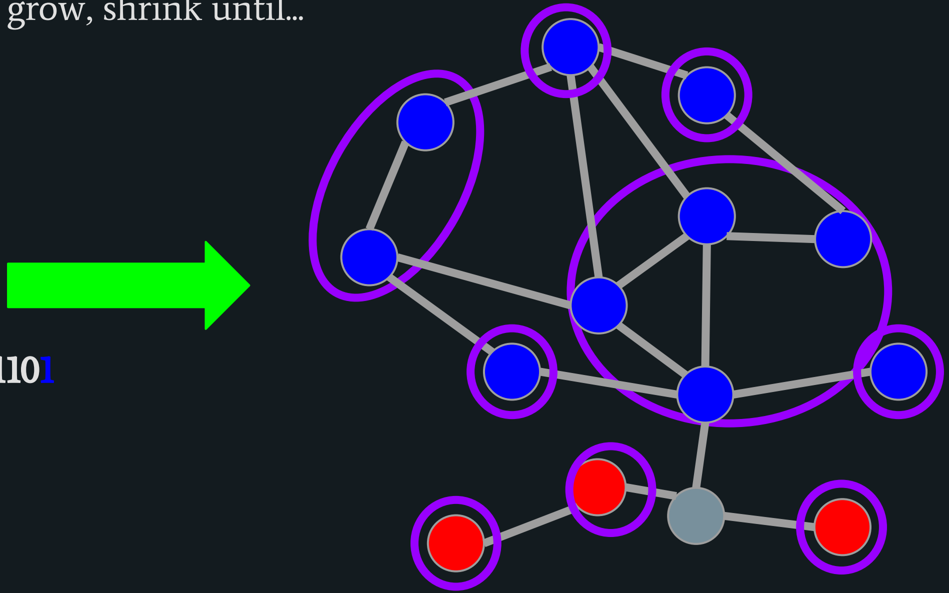
Network decomposition:



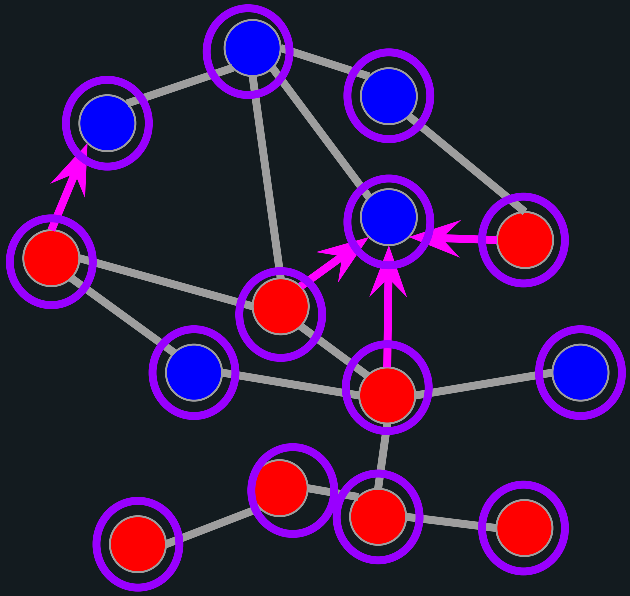
Network decomposition:



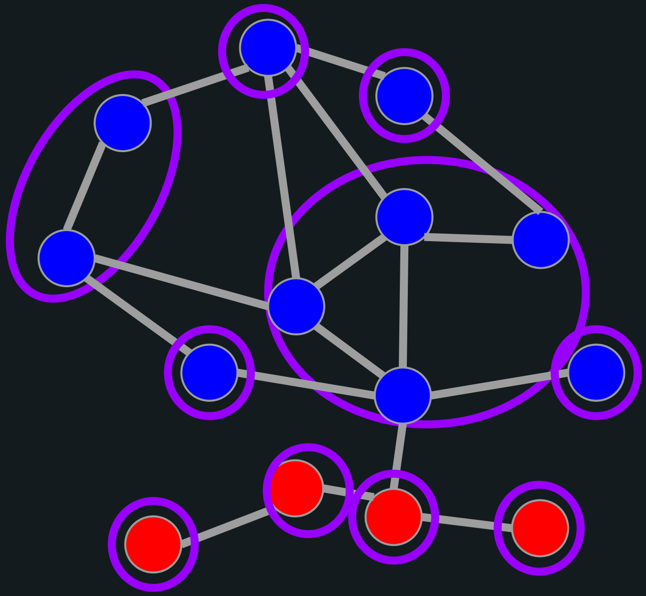
Clusters grow, shrink until...



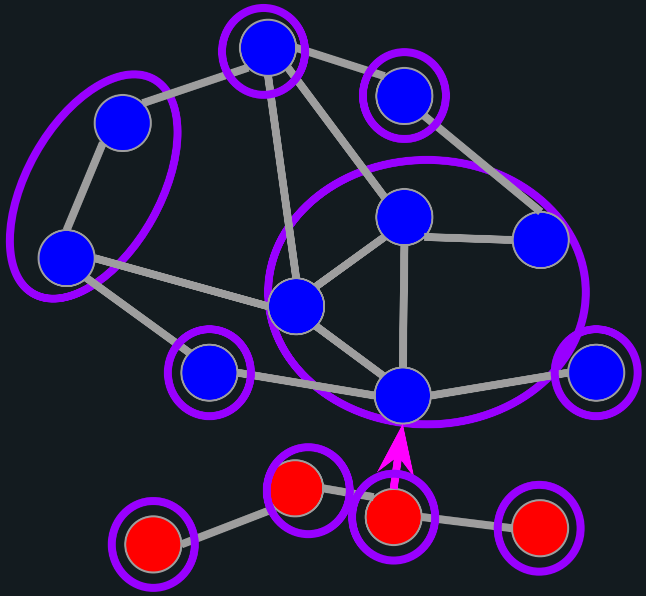
Network decomposition:



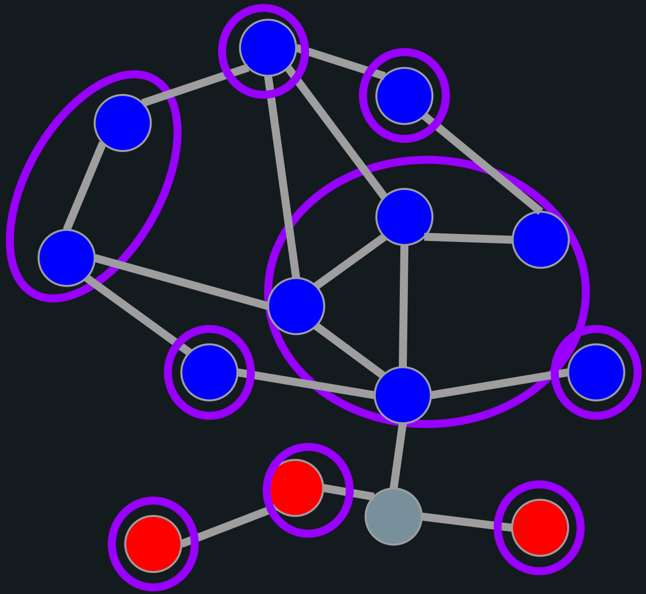
Network decomposition:



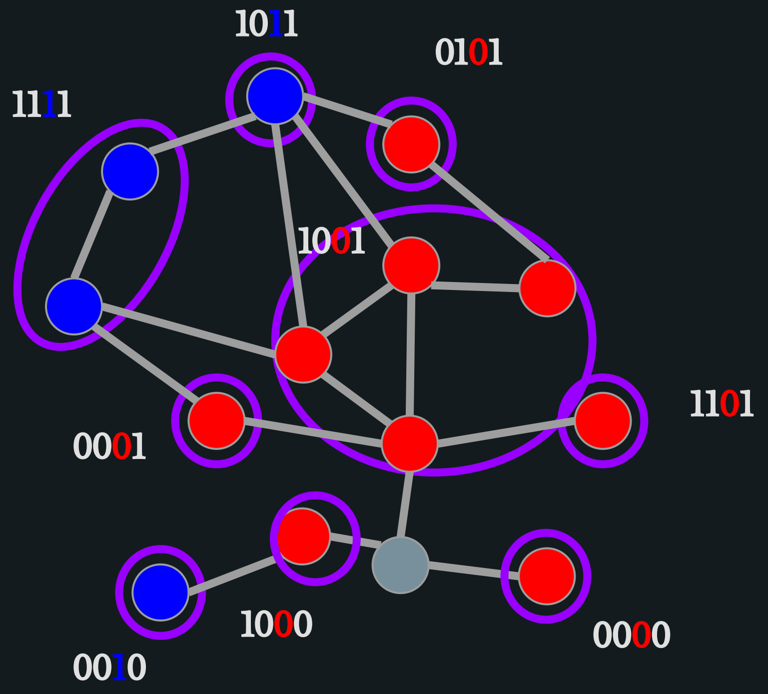
Network decomposition:



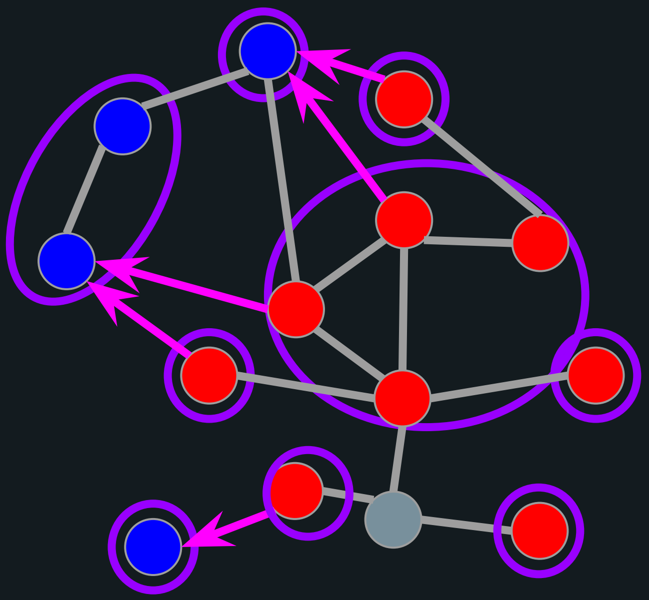
Network decomposition:



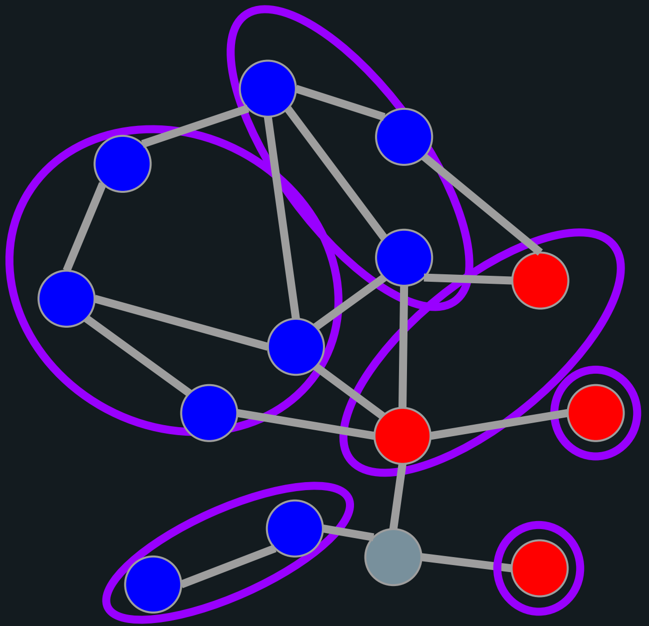
Network decomposition:



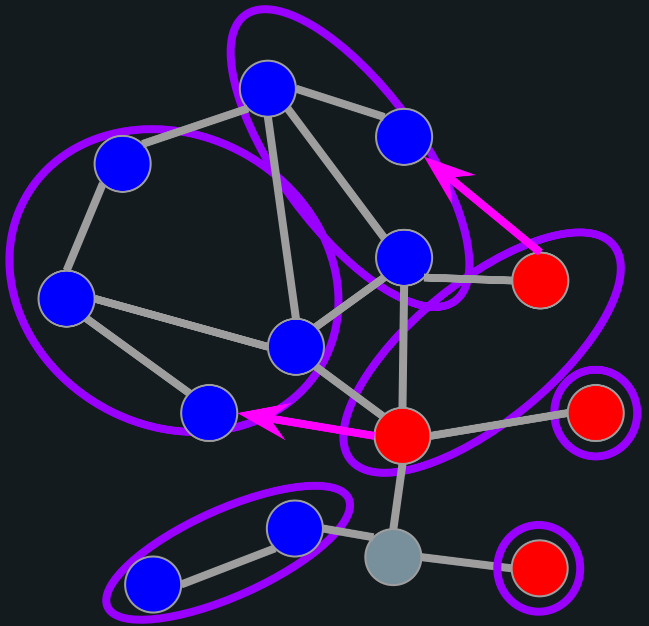
Network decomposition:



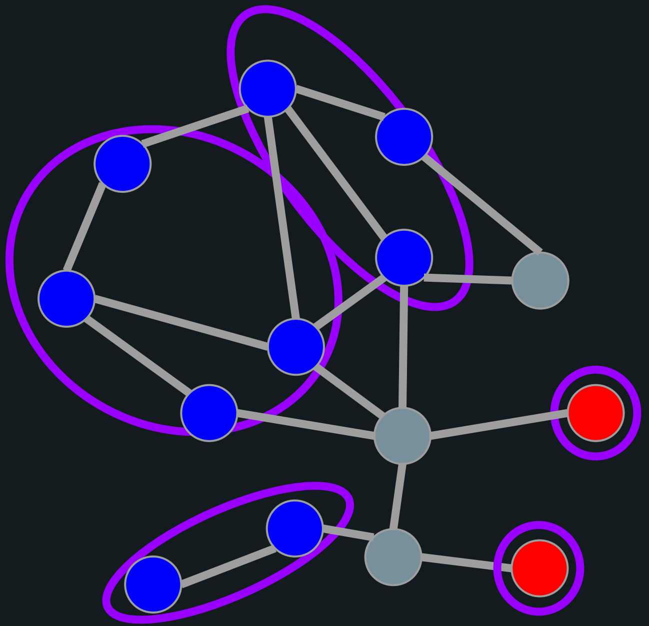
Network decomposition:



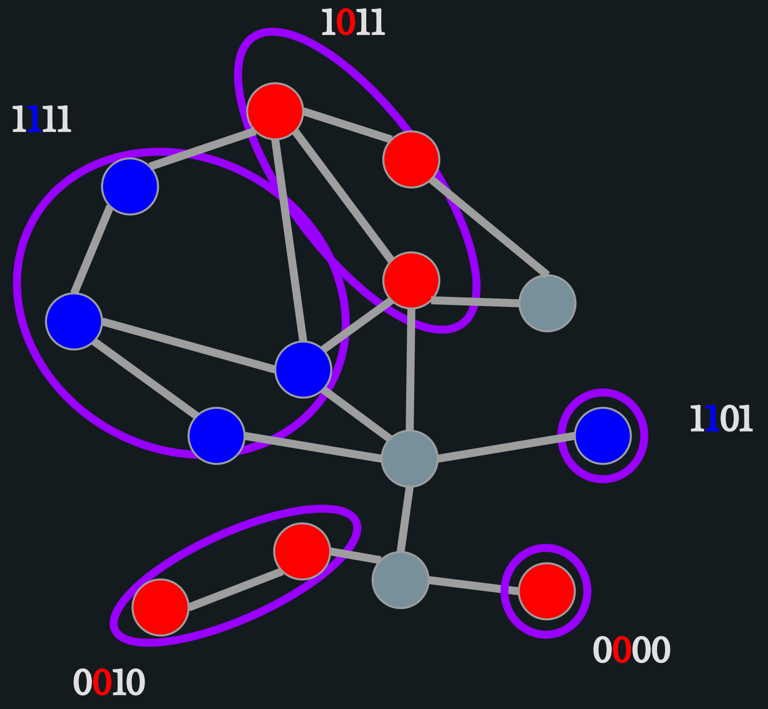
Network decomposition:



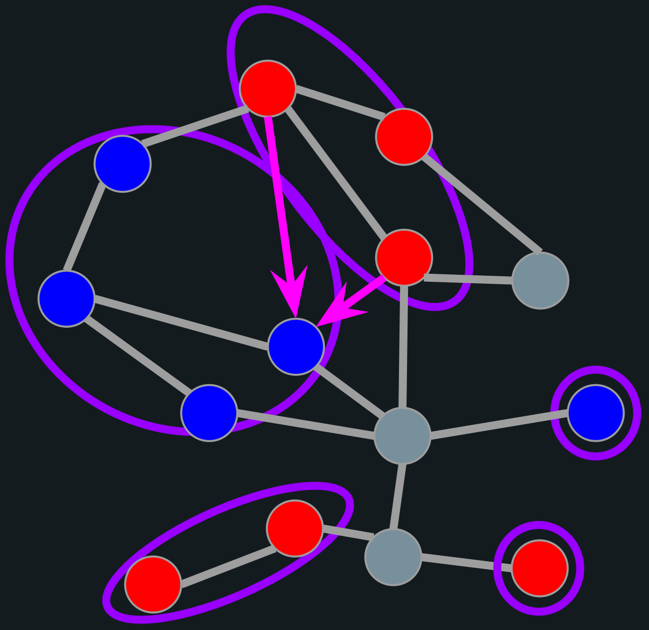
Network decomposition:



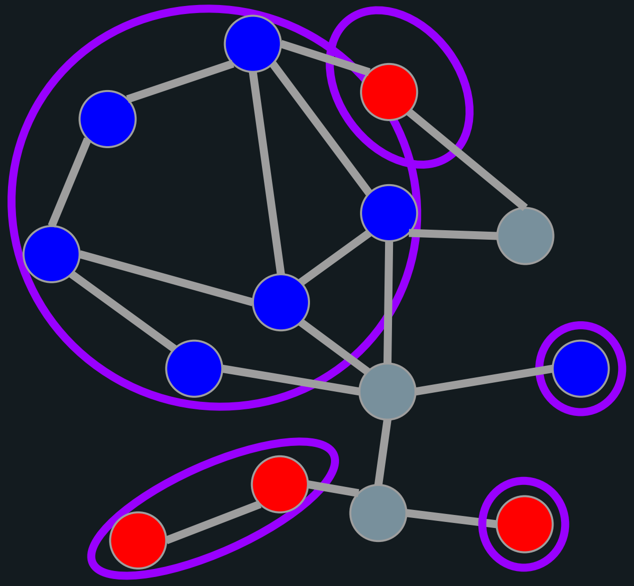
Network decomposition:



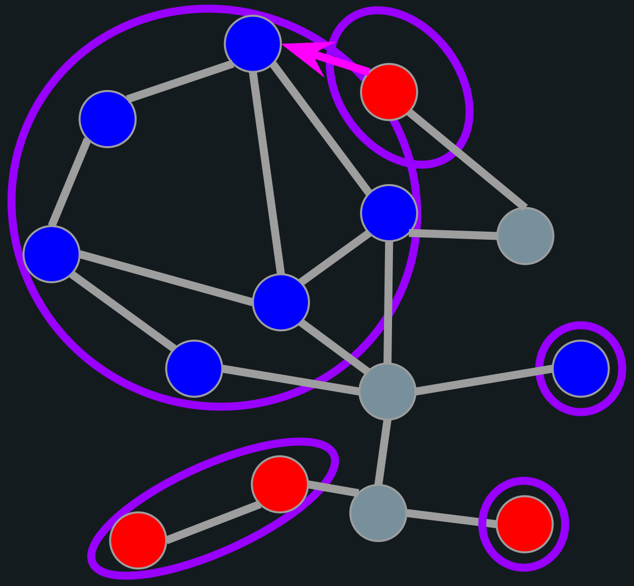
Network decomposition:



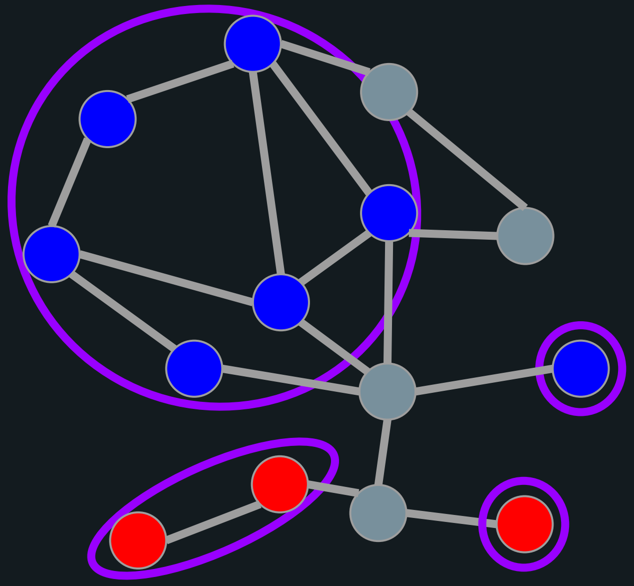
Network decomposition:



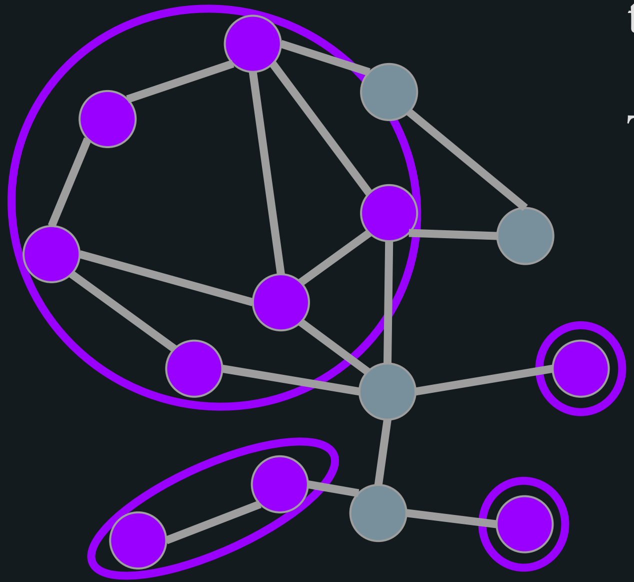
Network decomposition:



Network decomposition:



Network decomposition:



Since we have $O(\log n)$ phases in total, we set the threshold to $O(1/\log n)$.

This yields $O(\log^7 n)$ round algorithm.

Conclusion

We have seen very simple algorithm that was needed for lot of theory in the LOCAL model.

Reasonable problems can now be solved deterministically in poly(log n) rounds.

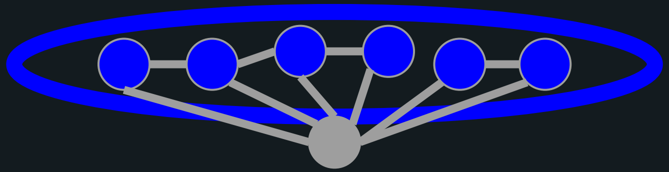
Many of the can be solved randomized in poly(log log n) rounds.

Outlook: CONGEST model

Since our algorithm works also in the CONGEST model, we get some more results:

[Censor-Hillel, Parter, Schwartzman DISC'17] + [R., Ghaffari 19+]:

“There is also deterministic O(poly(log n))-round algorithm for MIS in the CONGEST model. “



Beyond distributed models

[Ghaffari, Kuhn, Uitto '19+] + [Chang, Fischer, Ghaffari, Uitto, Zheng PODC'19]

To get faster randomized algorithm for $\Delta+1$-coloring in the MPC model, it is both **necessary** and **sufficient** to improve deterministic distributed algorithm for the same problem.