Faster $(\Delta + 1)$-Edge Coloring: Breaking the $mn^{1/2}$ Time Barrier

Sayan Bhattacharya[1], Din Carmon[2], Martín Costa[1], Shay Solomon[2], Tianyi Zhang[3]

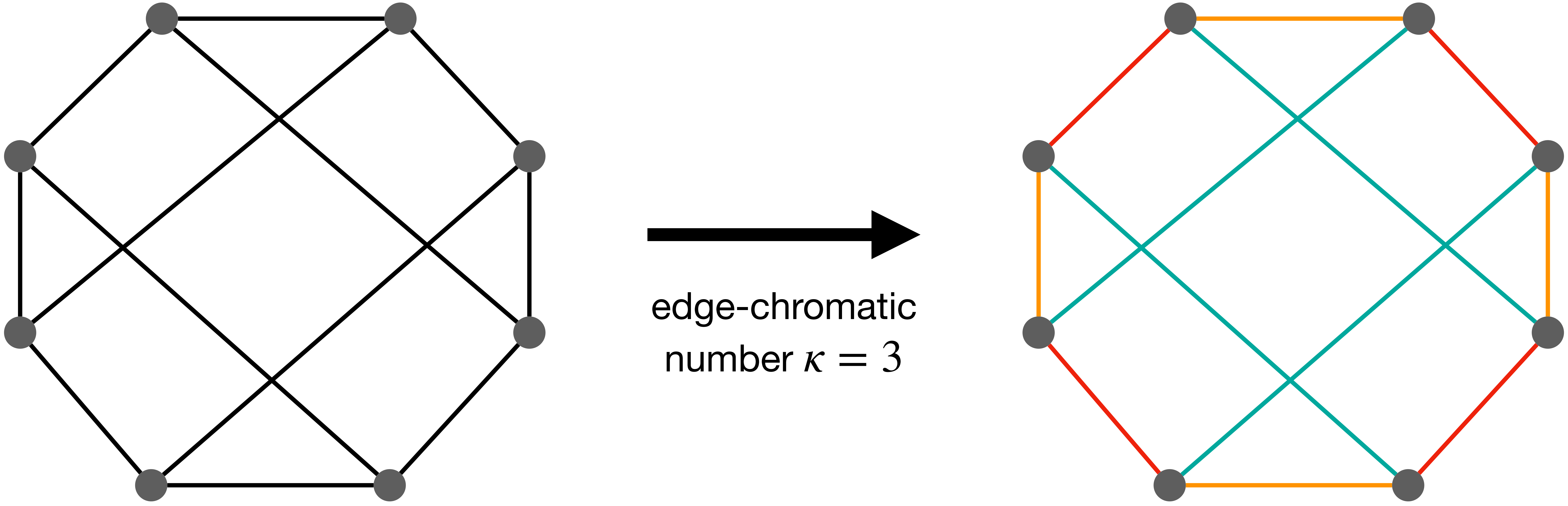
University of Warwick[1]

Tel Aviv University[2]

ETH Zürich[3]

Edge Coloring

- Given an undirected simple graph $G = (V, E)$
- Compute a coloring $\chi : E \rightarrow \{1,2,\ldots,\kappa\}$ s.t. adjacent edges have different colors



Edge Coloring

- Given an undirected simple graph $G = (V, E)$
- Compute a coloring $\chi : E \rightarrow \{1,2,\ldots,\kappa\}$ s.t. adjacent edges have different colors

Why is edge coloring **relevant** to this workshop?

- Studies in the **distributed** setting:
  [EPS16] [BEPS16] [FGK17] [GHK18] [CHL+19] [Bern22] [Chr23] …
- Studies in the **dynamic** setting:
  [BM17] [BCHN18] [DHZ19] [Chr23] [BCPS24] [Chr24]

Edge Coloring

- Given an undirected simple graph $G = (V, E)$
- Compute a coloring $\chi : E \to \{1,2,\ldots,\kappa\}$ s.t. adjacent edges have different colors. What is the **smallest possible κ** ?
- **Folklore:** $\kappa \geq \Delta(G)$, here $\Delta = \max\{\deg(v)\}$
- **Upper bounds:**
  $\kappa \leq \Delta$ in bipartite graphs, $\kappa \leq \Delta + 1$ in general graphs [Vizing, 1964]
- **Hardness:**
  NP-hard to decide $\kappa = \Delta$ or $\Delta + 1$ [Holyer, 1981]

Edge Coloring

Δ-edge coloring in bipartite graphs is in near-linear time e.g. in [COS, 2001]

Question: The exact runtime of $(\Delta + 1)$-edge coloring?

Edge Coloring

Δ-edge coloring in bipartite graphs is in near-linear time e.g. in [COS, 2001]

Question: The exact runtime of $(\Delta + 1)$-edge coloring?

A sparse history of runtime complexities: $n = |V|, m = |E|$

- $O(mn)$ [Vizing, Diskret. Analiz 1964]
- $\tilde{O}(\min\{mn^{1/2}, m\Delta\})$ [Arjomandi, INFOR 1982] [Gabow et al, 1985]
- $O(mn^{1/2})$ log-shaving [Sinnamon, 2019]

Edge Coloring

Δ-edge coloring in bipartite graphs is in near-linear time e.g. in [COS, 2001]

Question: The exact runtime of $(\Delta + 1)$-edge coloring?

A sparse history of runtime complexities: $n = |V|, m = |E|$

- $O(mn)$ [Vizing, Diskret. Analiz 1964]
- $\tilde{O}(\min\{mn^{1/2}, m\Delta\})$ [Arjomandi, INFOR 1982] [Gabow et al, 1985]
- $O(mn^{1/2})$ log-shaving [Sinnamon, 2019]
- $\tilde{O}(mn^{1/3})$ [BCCSZ, FOCS 2024]
- $\tilde{O}(n^2)$ **concurrent** [Assadi, SODA 2025]

Edge Coloring

Δ-edge coloring in bipartite graphs is in near-linear time e.g. in [COS, 2001]

Question: The exact runtime of $(\Delta + 1)$-edge coloring?

A sparse history of runtime complexities: $n = |V|, m = |E|$

- $O(mn)$ [Vizing, Diskret. Analiz 1964]
- $\tilde{O}(\min\{mn^{1/2}, m\Delta\})$ [Arjomandi, INFOR 1982] [Gabow et al, 1985]
- $O(mn^{1/2})$ log-shaving [Sinnamon, 2019]
- $\tilde{O}(mn^{1/3})$ [BCCSZ, FOCS 2024]
- $\tilde{O}(n^2)$ **concurrent** [Assadi, SODA 2025]

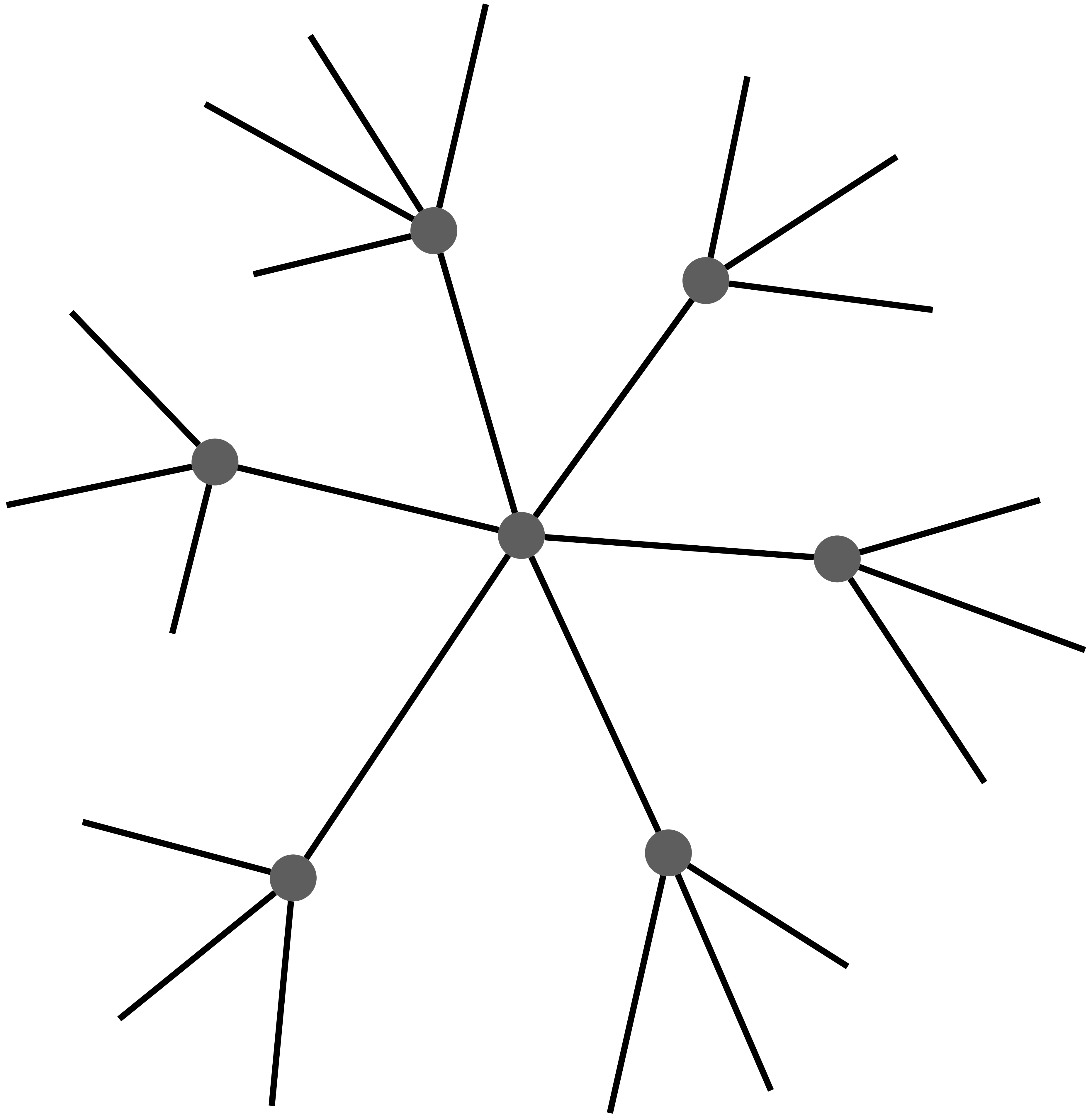
Previous Bottleneck

Reduce to $O(m/\Delta) = O(n)$ edges

1. Divide $G = G_1 \cup G_2$ such that $\Delta(G_1), \Delta(G_2) \le \Delta/2$

2. Color G_1, G_2 recursively using $2 \cdot (\Delta/2 + 1) = \Delta + 2$ colors

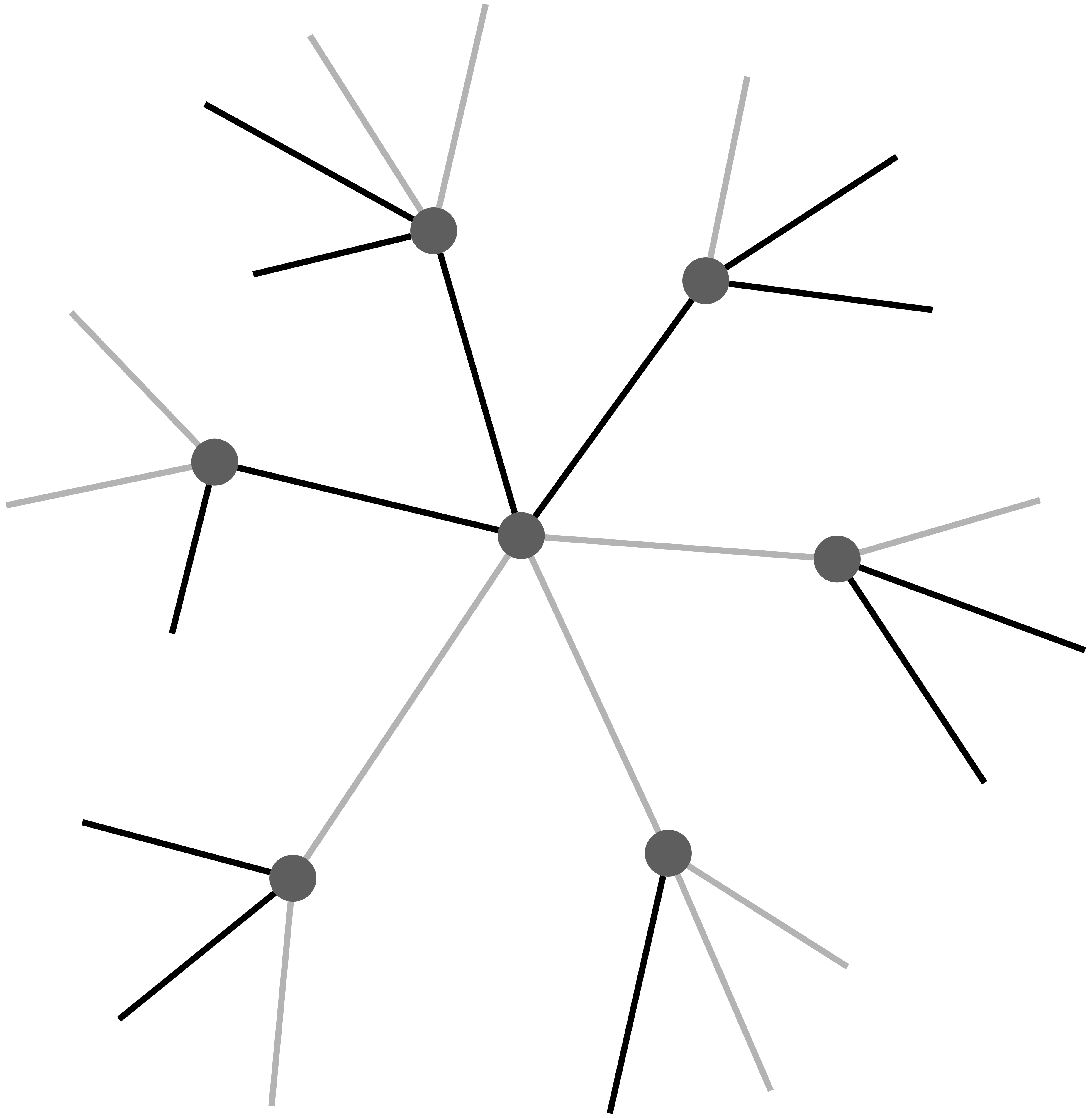
3. Delete the least-popular color, $O(m/\Delta) = O(n)$ uncolored edges



Previous Bottleneck

Reduce to $O(m/\Delta) = O(n)$ edges

1. **Divide $G = G_1 \cup G_2$ such that $\Delta(G_1), \Delta(G_2) \leq \Delta/2$**
2. Color G_1, G_2 recursively using $2 \cdot (\Delta/2 + 1) = \Delta + 2$ colors
3. Delete the least-popular color, $O(m/\Delta) = O(n)$ uncolored edges



Previous Bottleneck

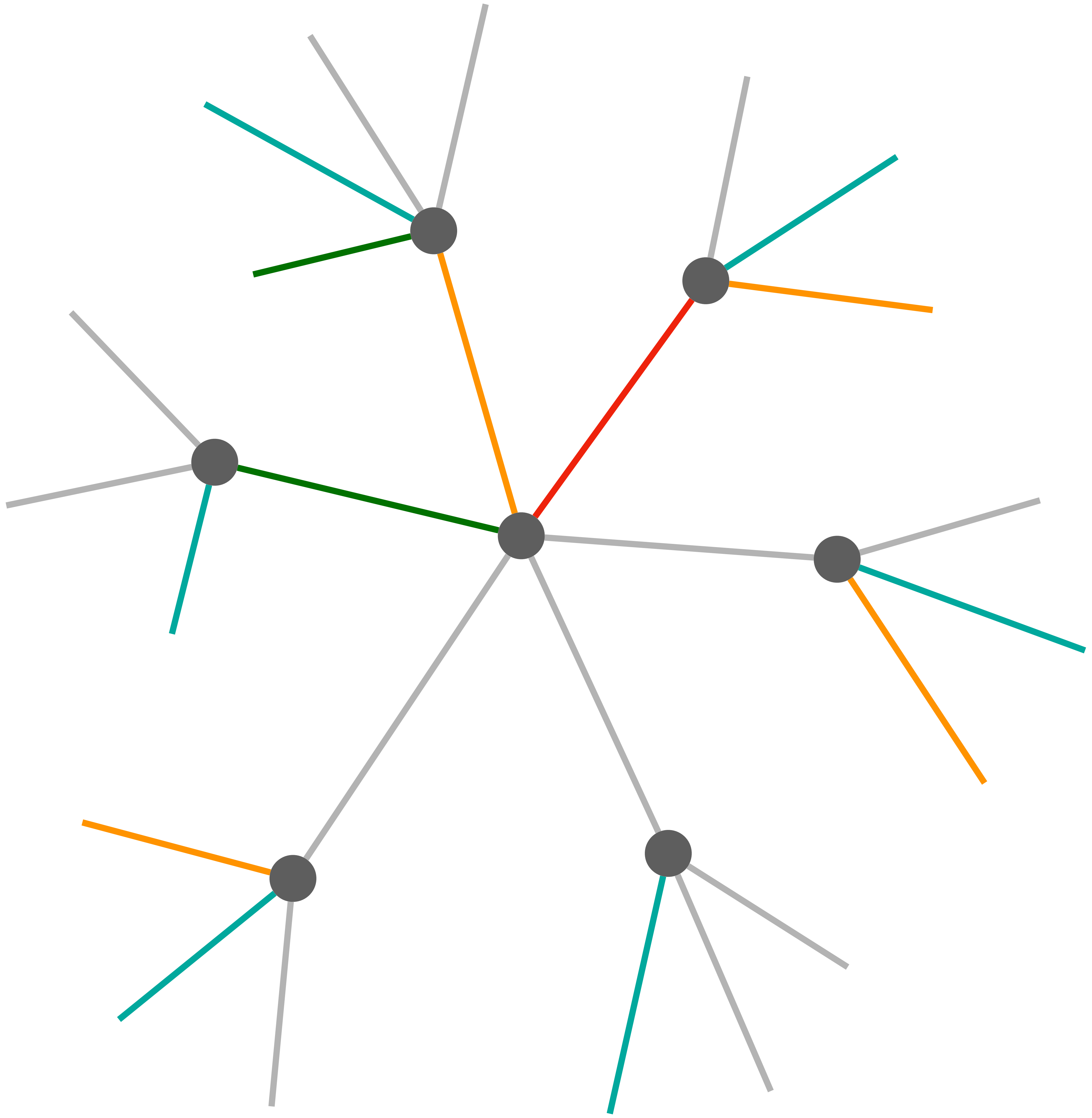
G_1 uses {■■■■}

Reduce to $O(m/\Delta) = O(n)$ edges

1. Divide $G = G_1 \cup G_2$ such that $\Delta(G_1), \Delta(G_2) \leq \Delta/2$

2. **Color G_1, G_2 recursively using $2 \cdot (\Delta/2 + 1) = \Delta + 2$ colors**

3. Delete the least-popular color, $O(m/\Delta) = O(n)$ uncolored edges



Previous Bottleneck

G_1 uses {}

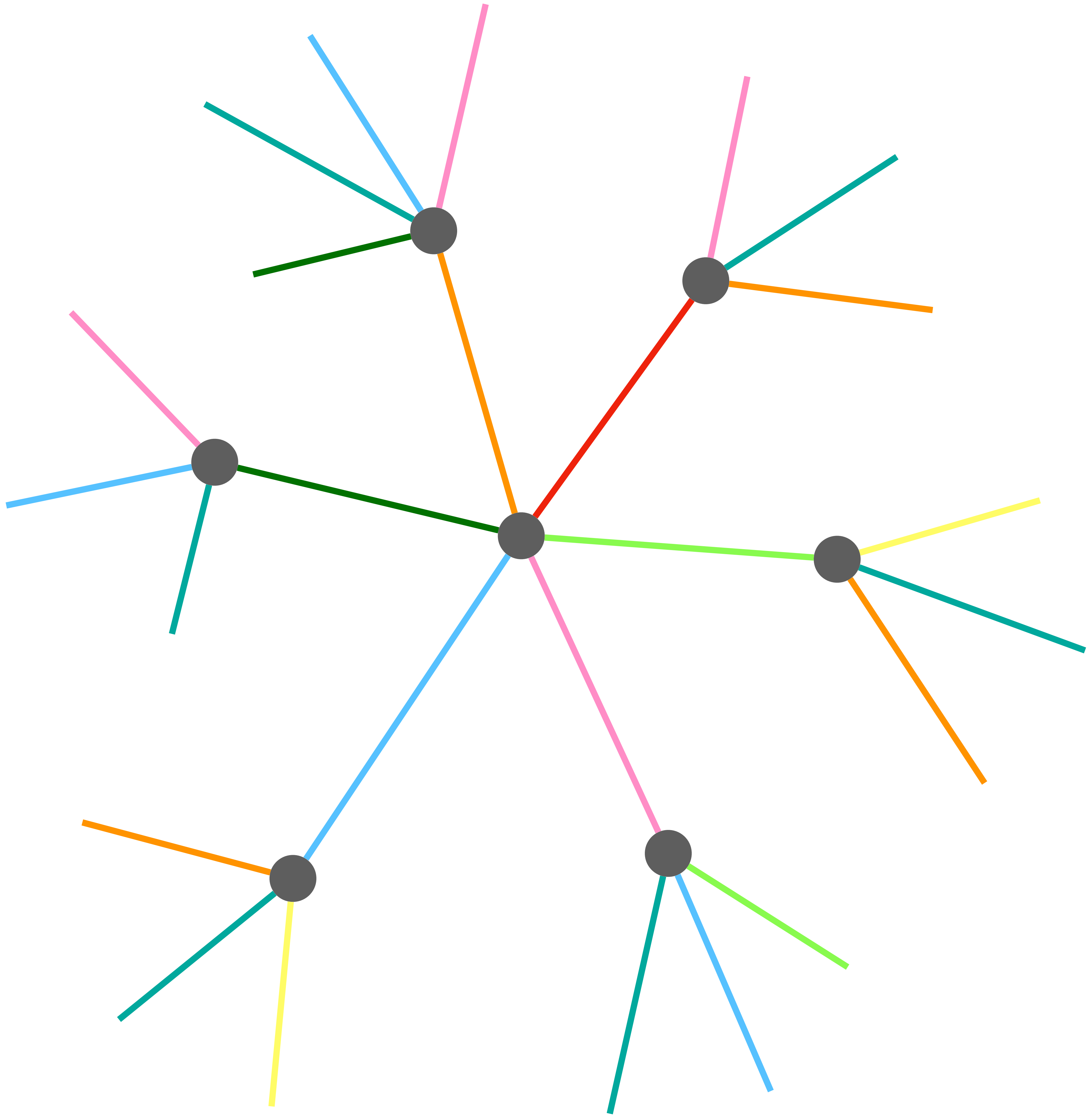
G_2 uses {}

Reduce to $O(m/\Delta) = O(n)$ edges

1. Divide $G = G_1 \cup G_2$ such that $\Delta(G_1), \Delta(G_2) \leq \Delta/2$

2. **Color G_1, G_2 recursively using $2 \cdot (\Delta/2 + 1) = \Delta + 2$ colors**

3. Delete the least-popular color, $O(m/\Delta) = O(n)$ uncolored edges



Previous Bottleneck

G_1 uses {}

G_2 uses {}

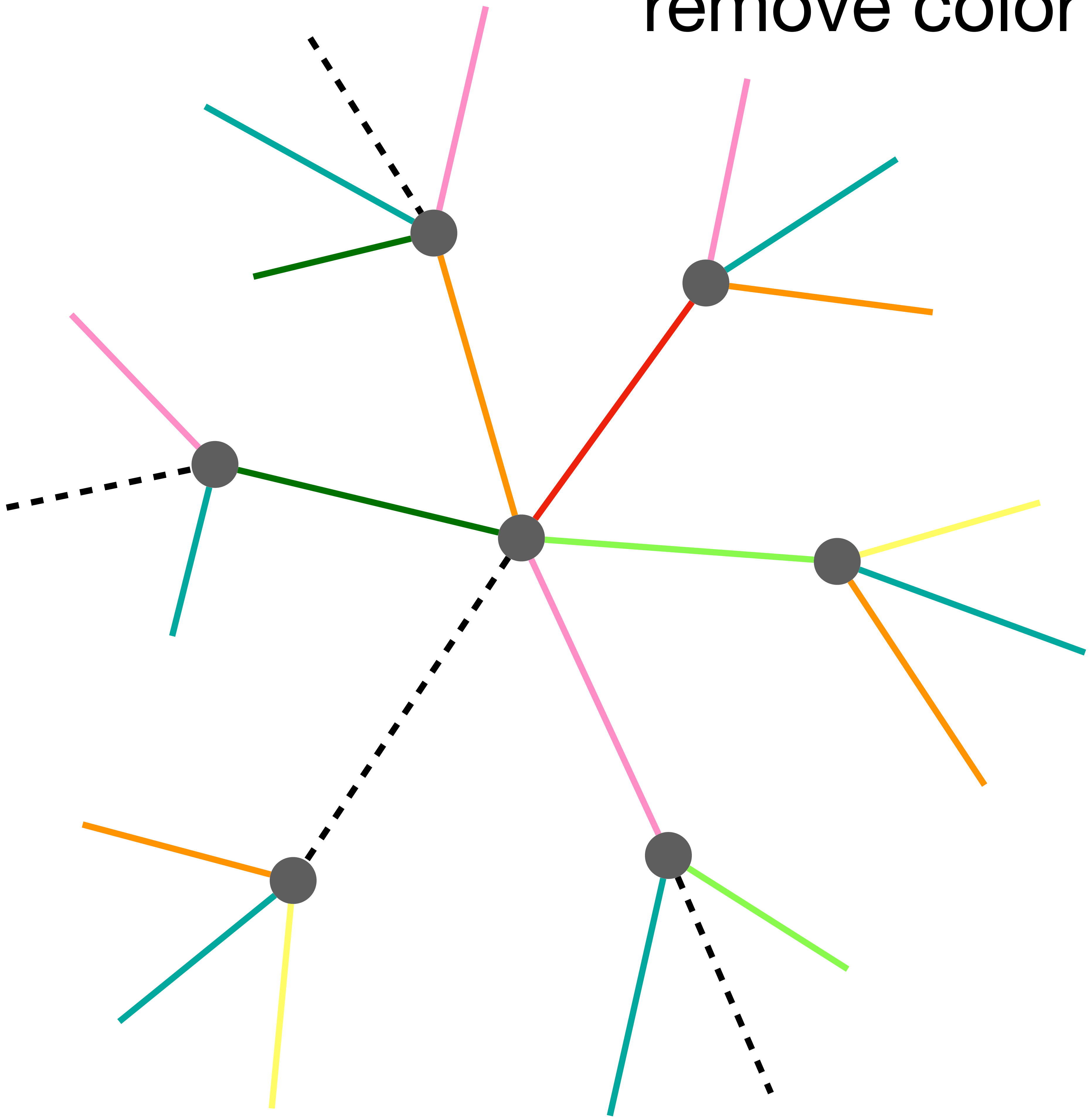
remove color

Reduce to $O(m/\Delta) = O(n)$ edges

1. Divide $G = G_1 \cup G_2$ such that $\Delta(G_1), \Delta(G_2) \leq \Delta/2$

2. Color G_1, G_2 recursively using $2 \cdot (\Delta/2 + 1) = \Delta + 2$ colors

3. **Delete the least-popular color, $O(m/\Delta) = O(n)$ uncolored edges**



Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), Vizing fans/chains (general)

Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

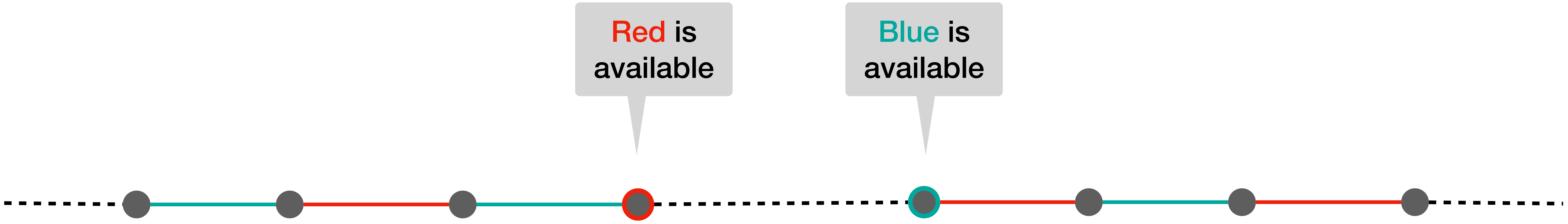
Answer: **Flip alternating paths (bipartite)**, Vizing fans/chains (general)

Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: **Flip alternating paths (bipartite)**, Vizing fans/chains (general)

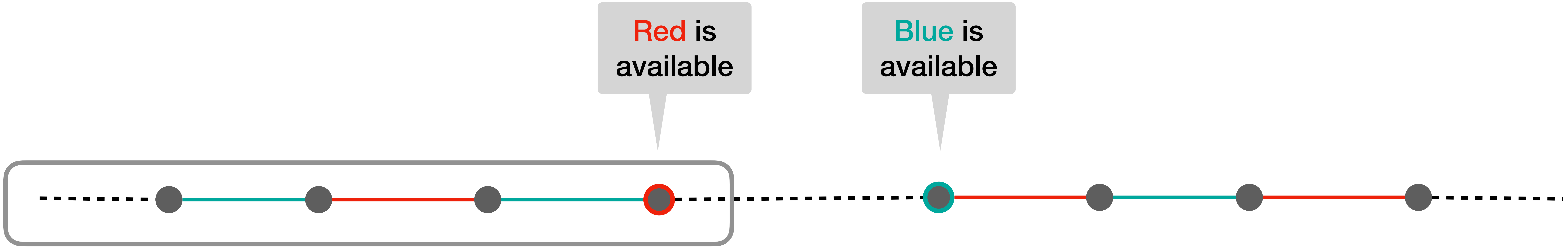


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), Vizing fans/chains (general)

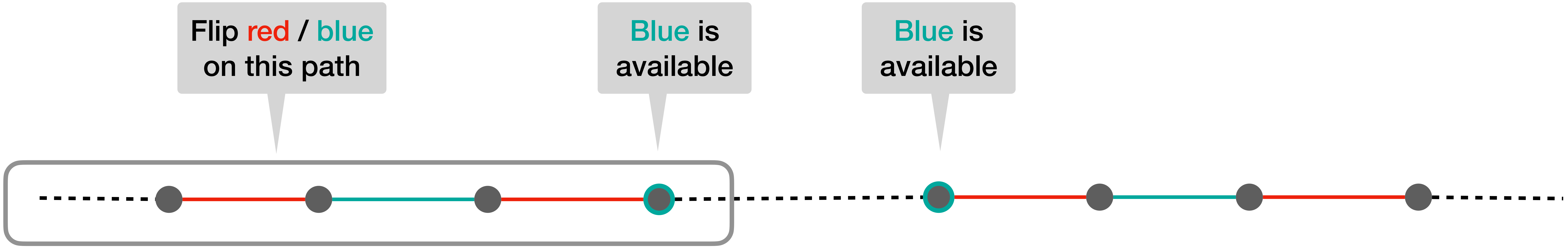


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: **Flip alternating paths (bipartite)**, Vizing fans/chains (general)



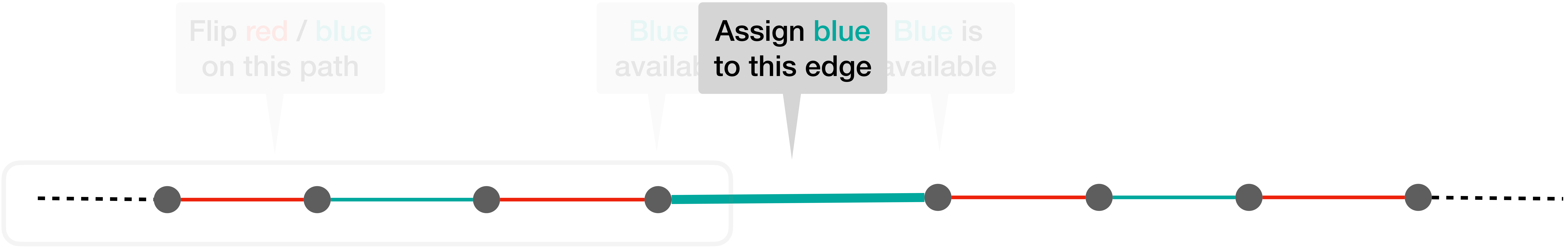
Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: **Flip alternating paths (bipartite)**, Vizing fans/chains (general)

Exercise: Where did we use bipartite-ness?

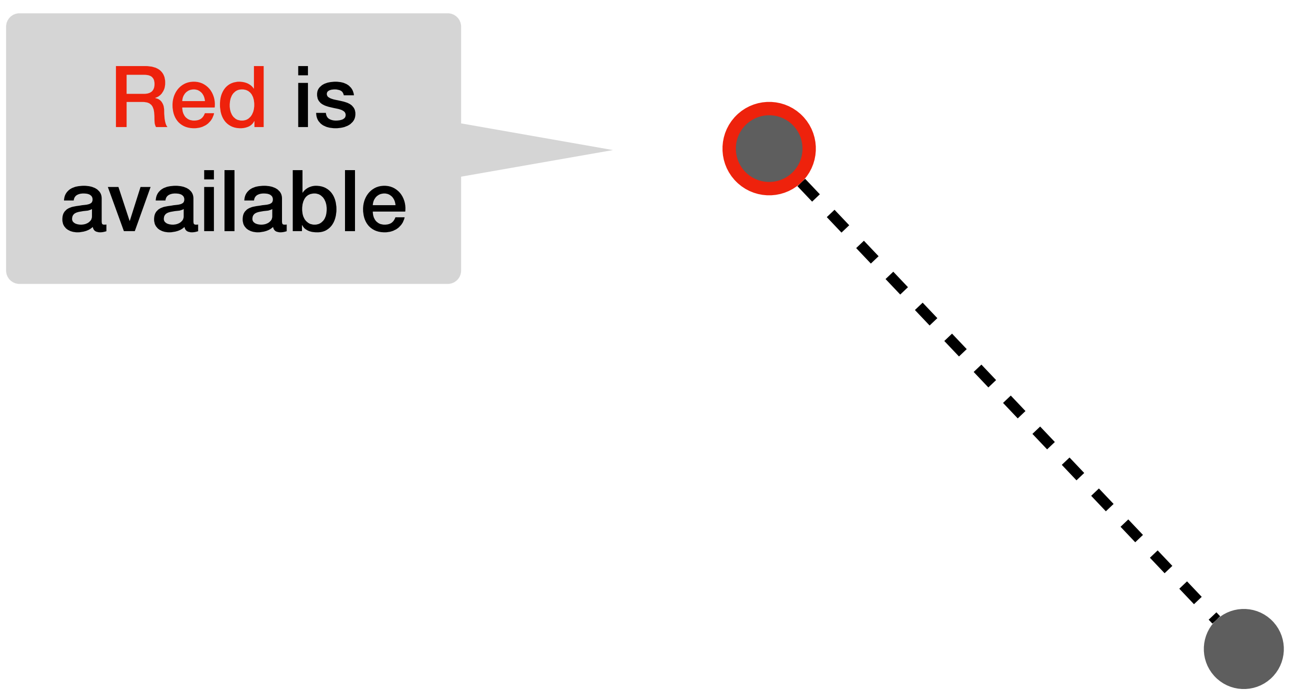


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), **Vizing fans/chains (general)**

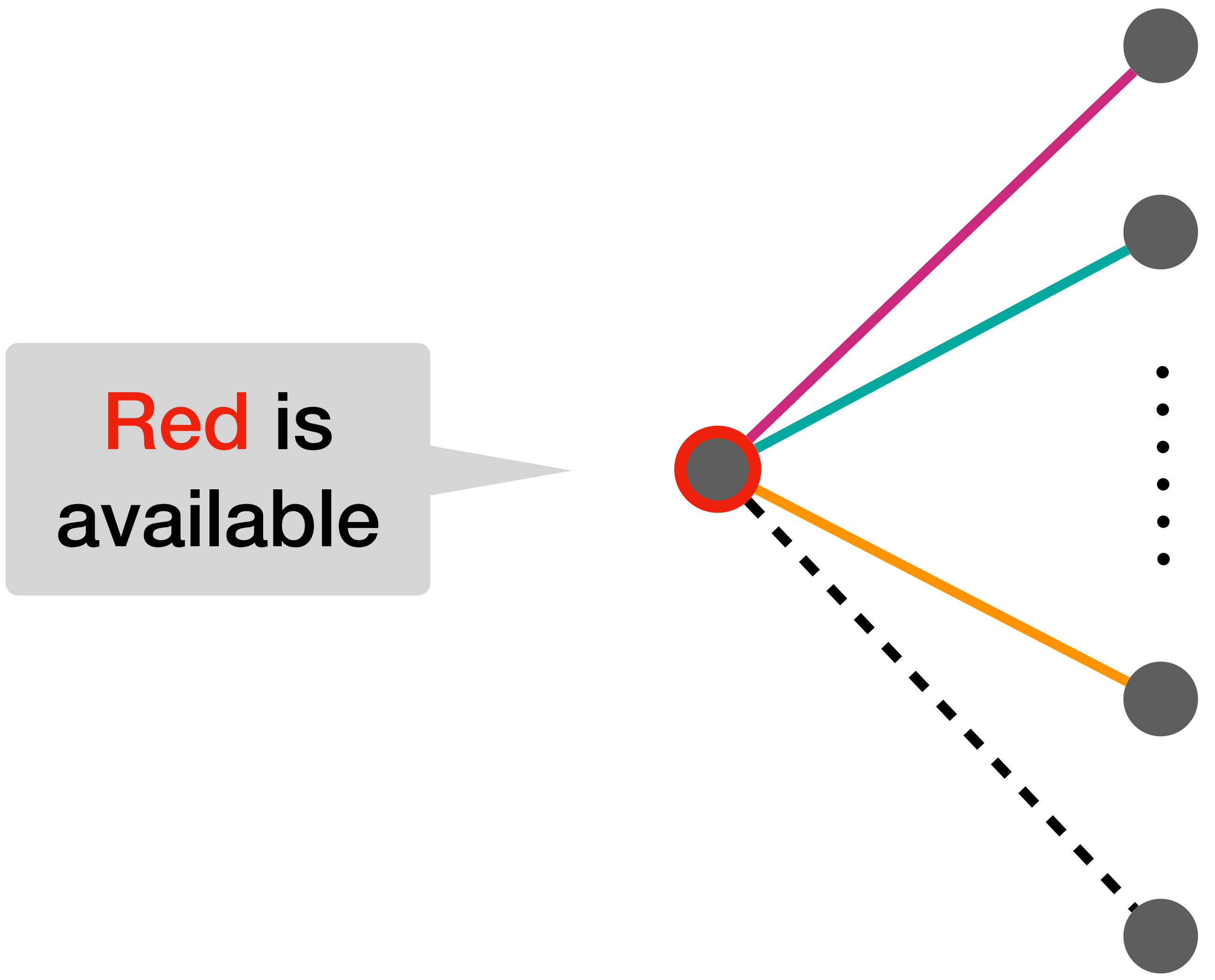


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), **Vizing fans/chains (general)**

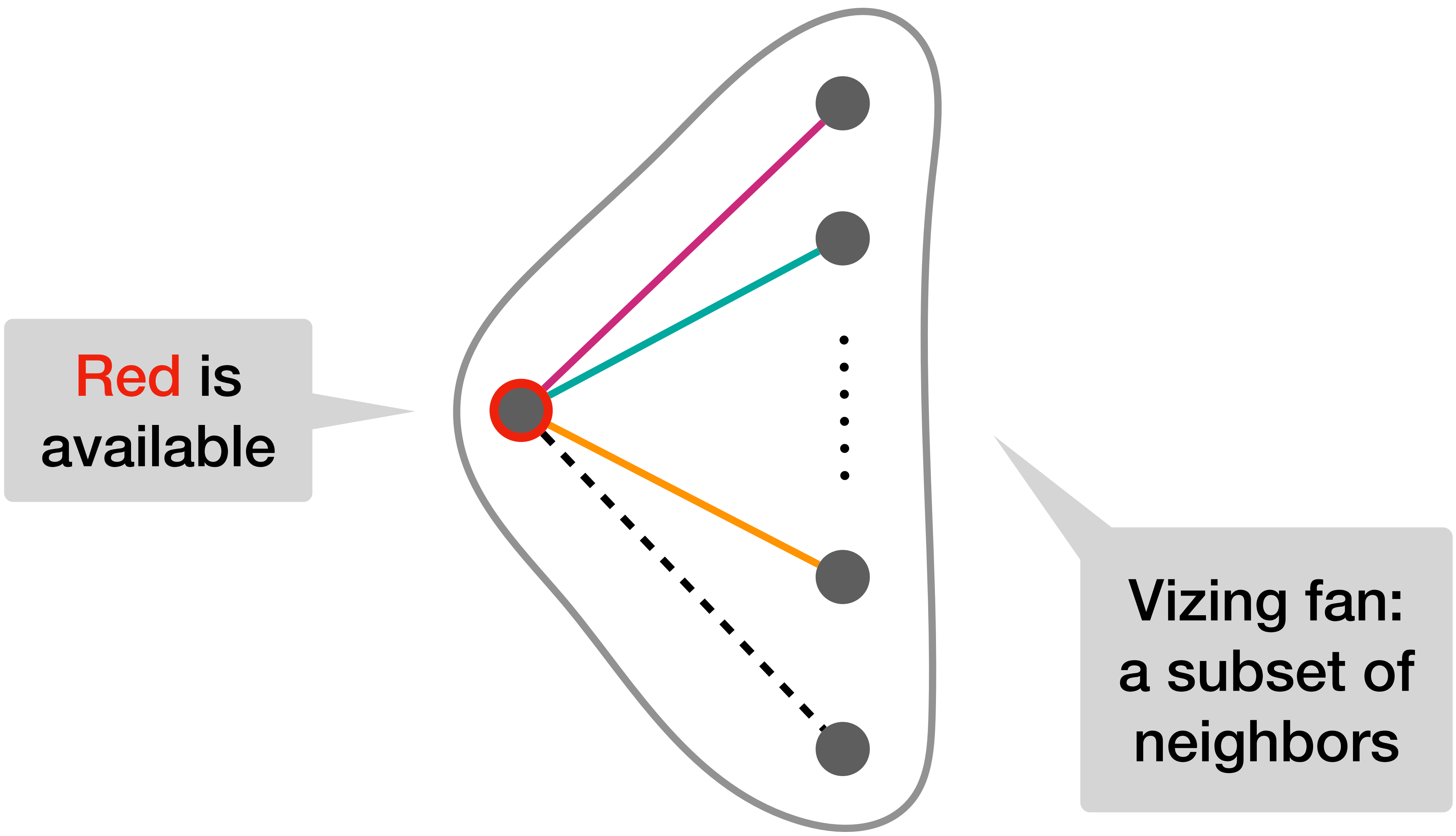


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), **Vizing fans/chains (general)**

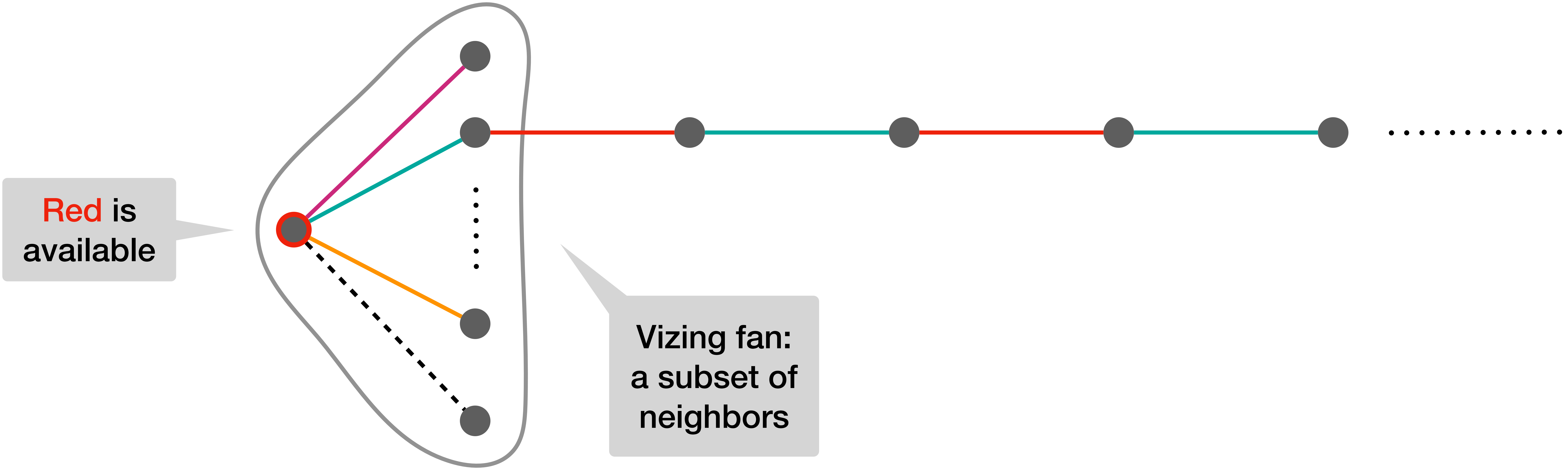


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), **Vizing fans/chains (general)**

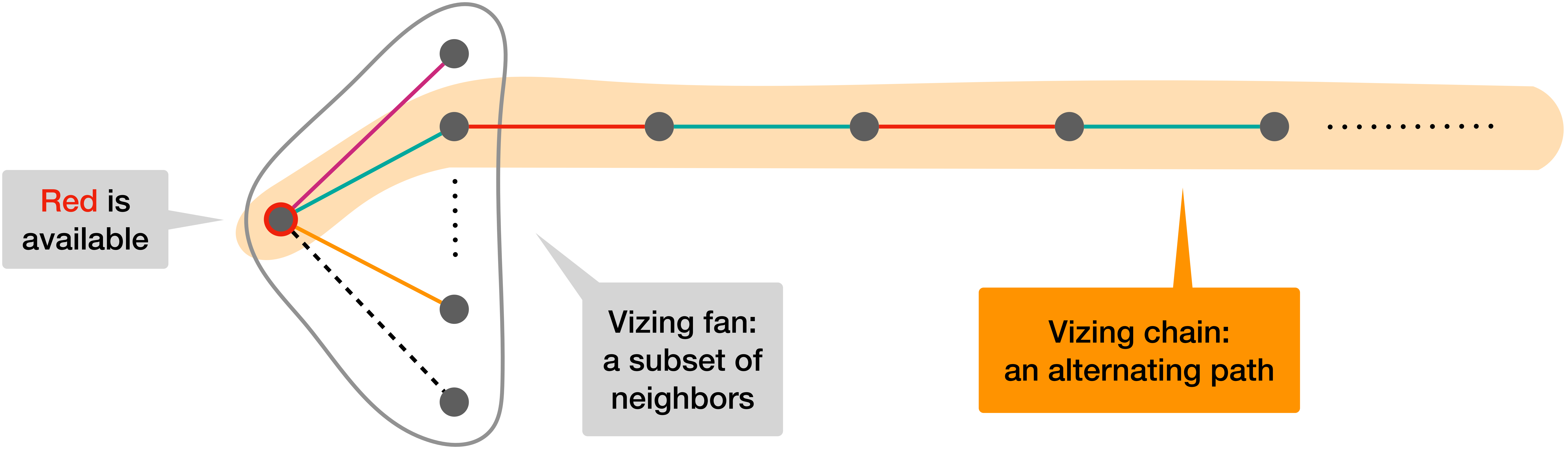


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Subtask: How to extend to a single uncolored edge?

Answer: Flip alternating paths (bipartite), **Vizing fans/chains (general)**

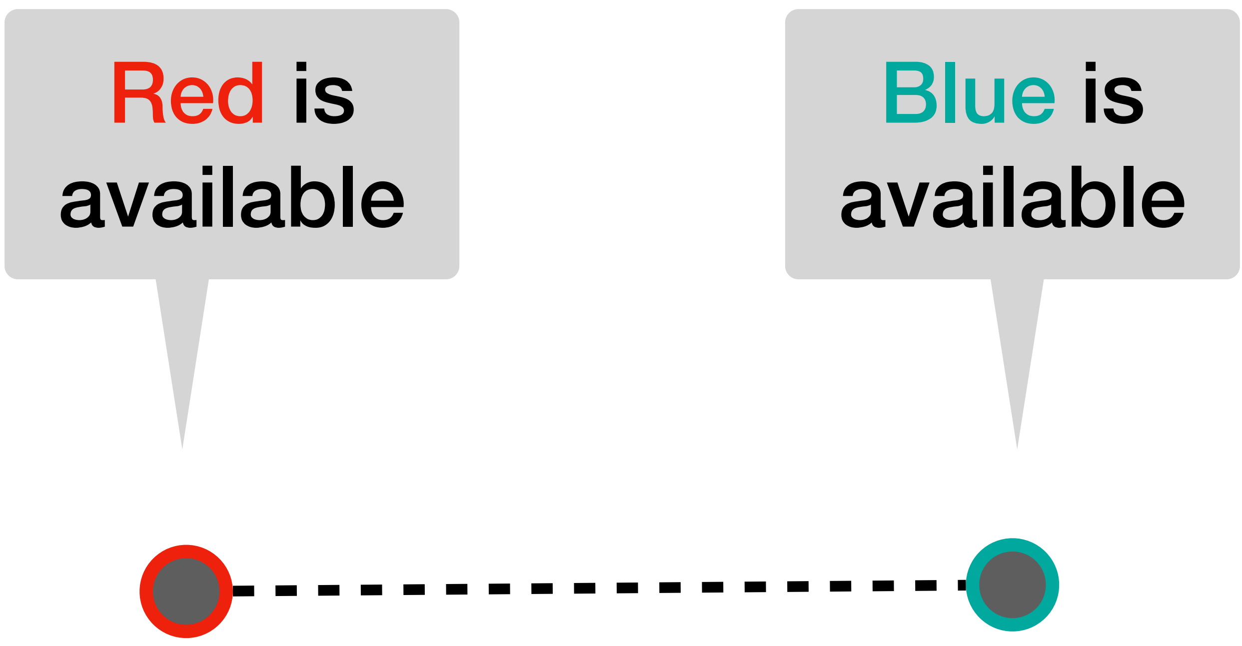


Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Runtime of color extension = **length** of alternating path

Technique: Analyze the average length of alternating paths



Call it a **type**-{**red**, **blue**} edge

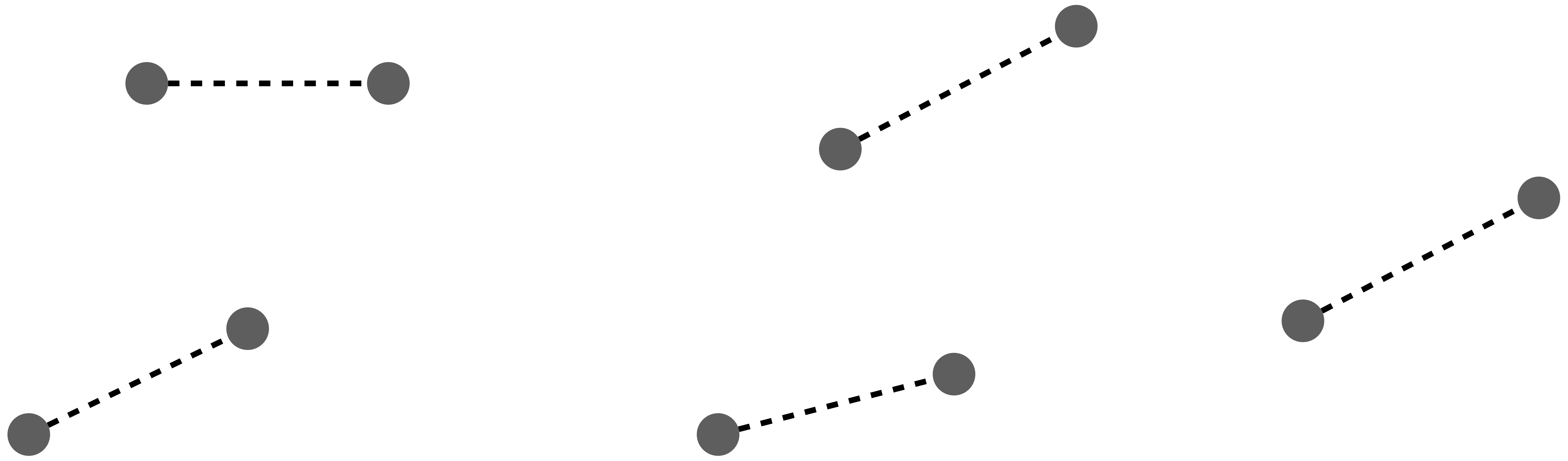
All edges are sorted into Δ^2 **types**

Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Runtime of color extension = **length** of alternating path

Technique: Analyze the average length of alternating paths



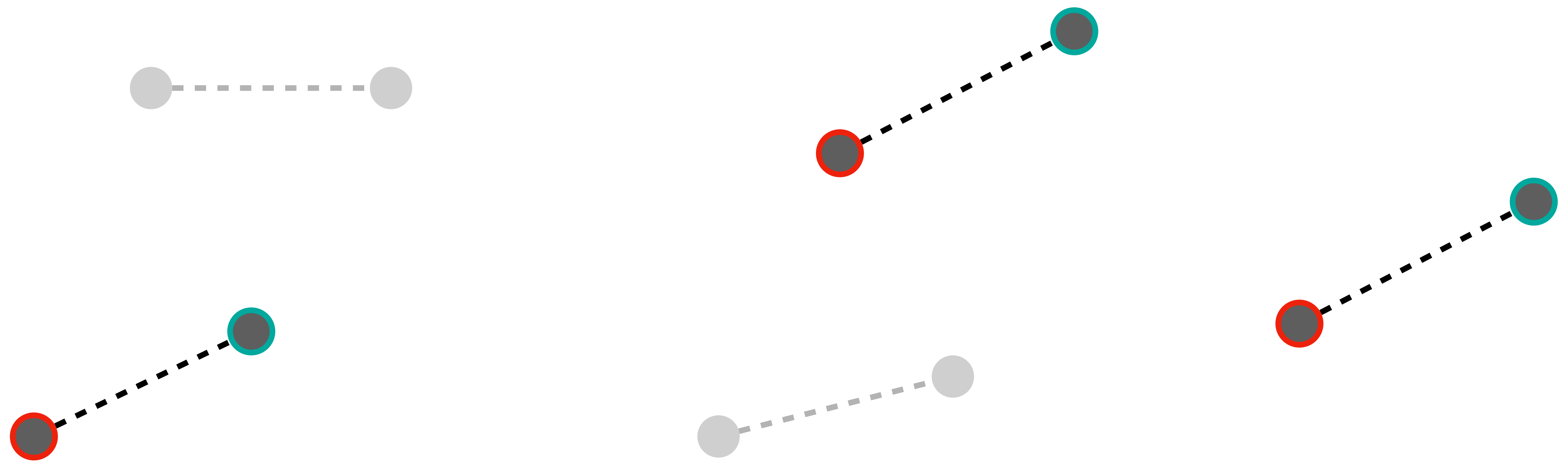
Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Runtime of color extension = **length** of alternating path

Technique: Analyze the average length of alternating paths

The an **average** type contains $\geq n/\Delta^2$ edges



Previous Bottleneck

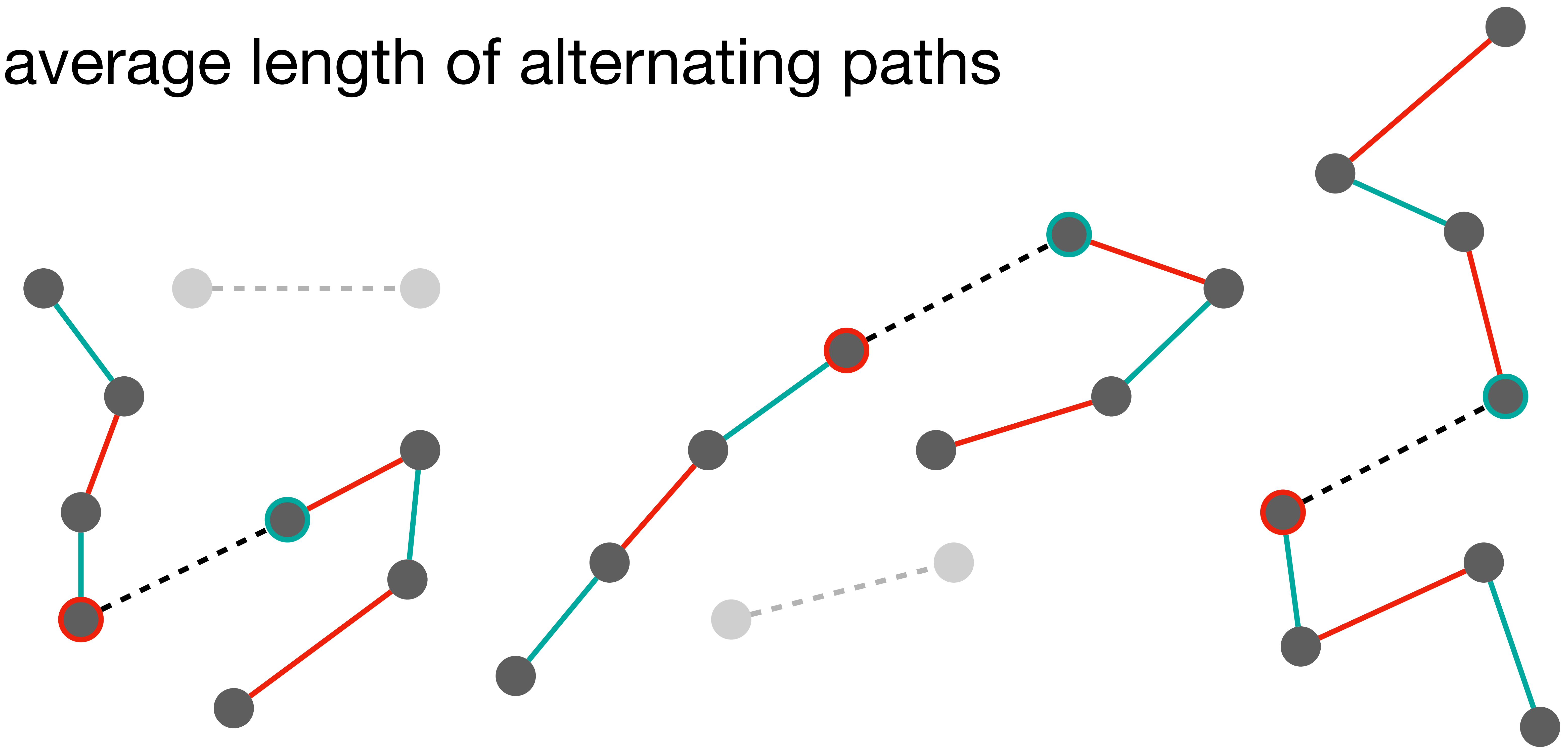
Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

Runtime of color extension = **length** of alternating path

Technique: Analyze the average length of alternating paths

The an **average** type contains $\geq n/\Delta^2$ edges

Alt-paths are **edge-disjoint**, ave-len $\leq \Delta^2$



Previous Bottleneck

Main task: Extend a partial $(\Delta + 1)$-coloring to the last $O(n)$ edges

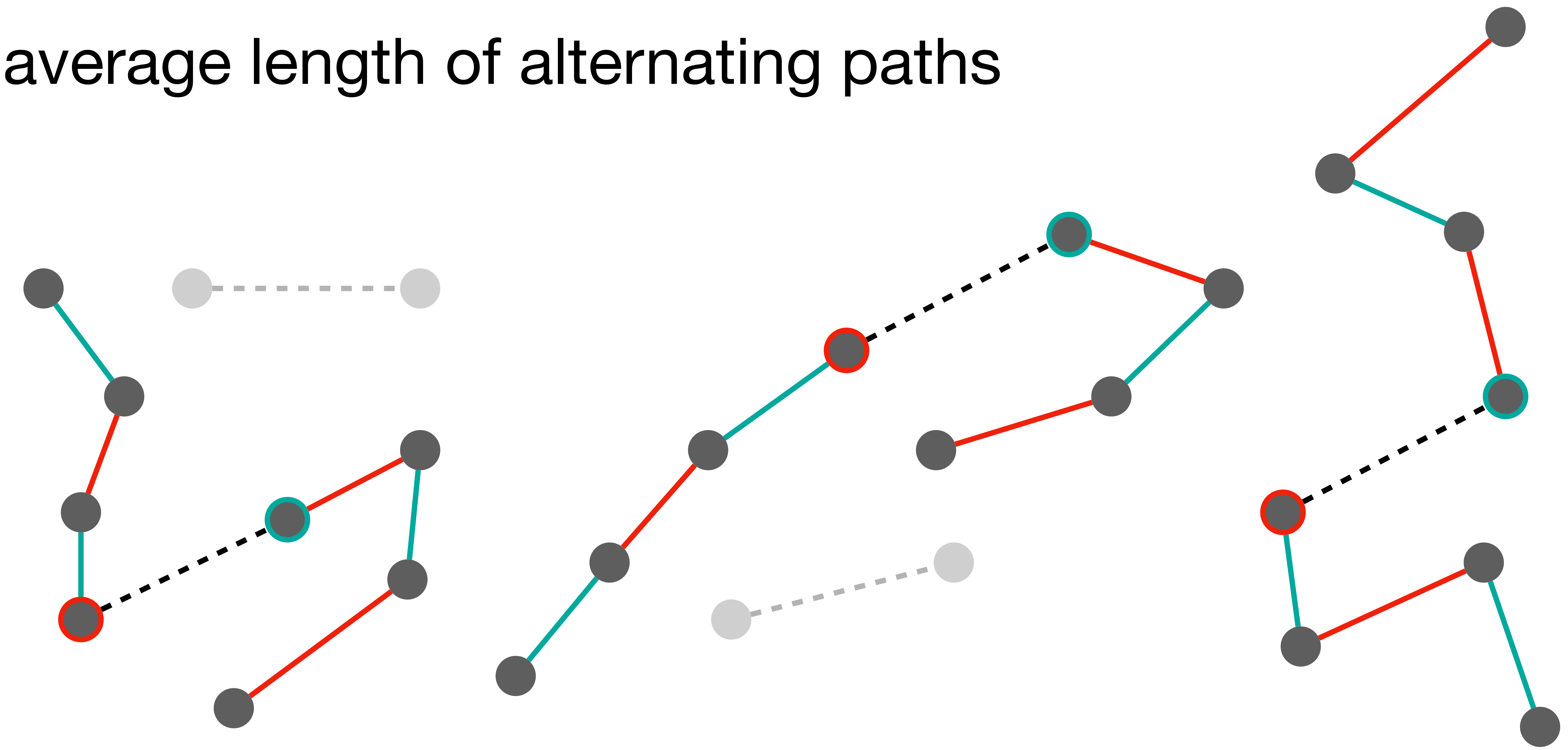
Runtime of color extension = **length** of alternating path

Technique: Analyze the average length of alternating paths

The an **average** type contains $\geq n/\Delta^2$ edges

Alt-paths are **edge-disjoint**, ave-len $\leq \Delta^2$

Coloring $O(n)$ edges takes $O(\Delta^2 n)$ time

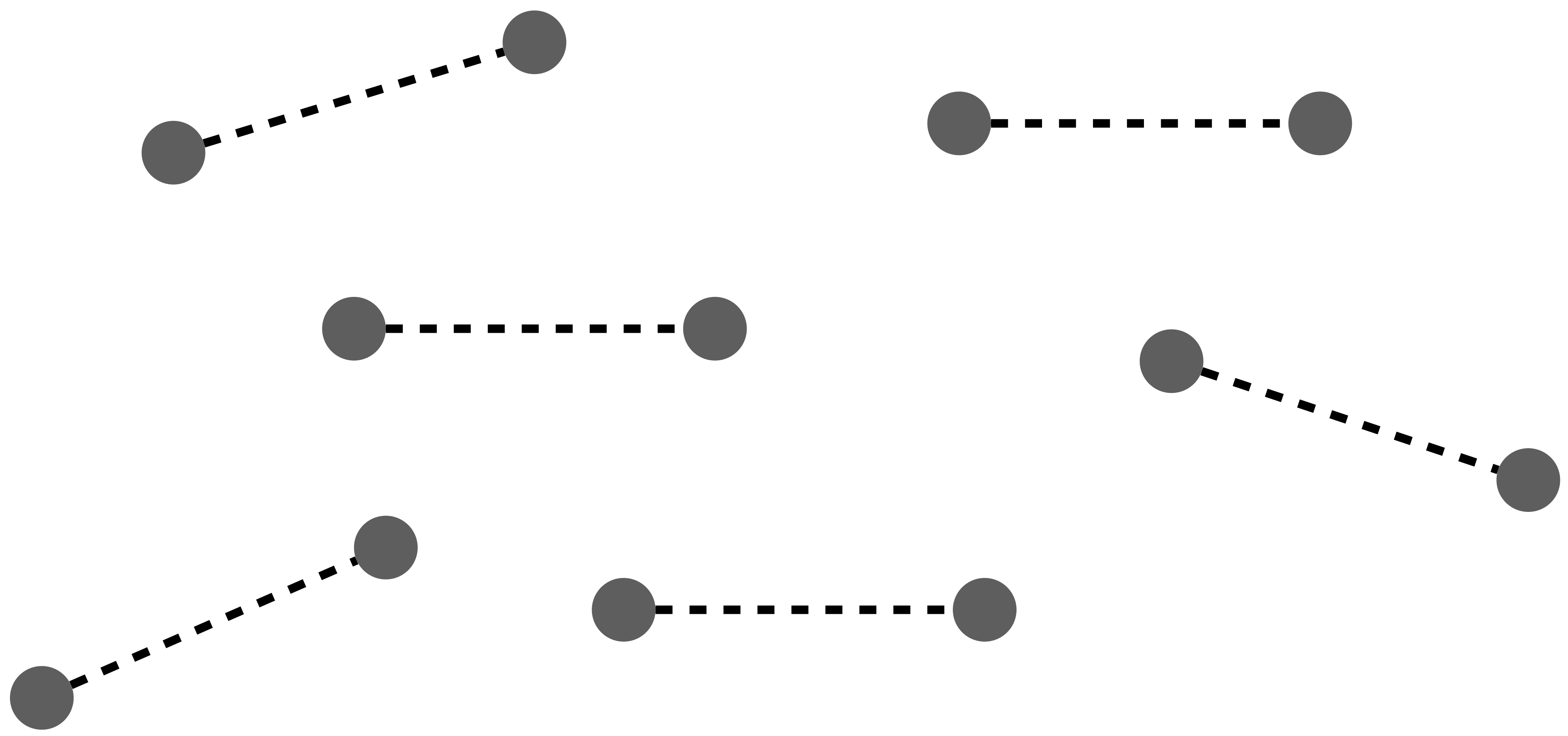


New Approach: Coloring Stars

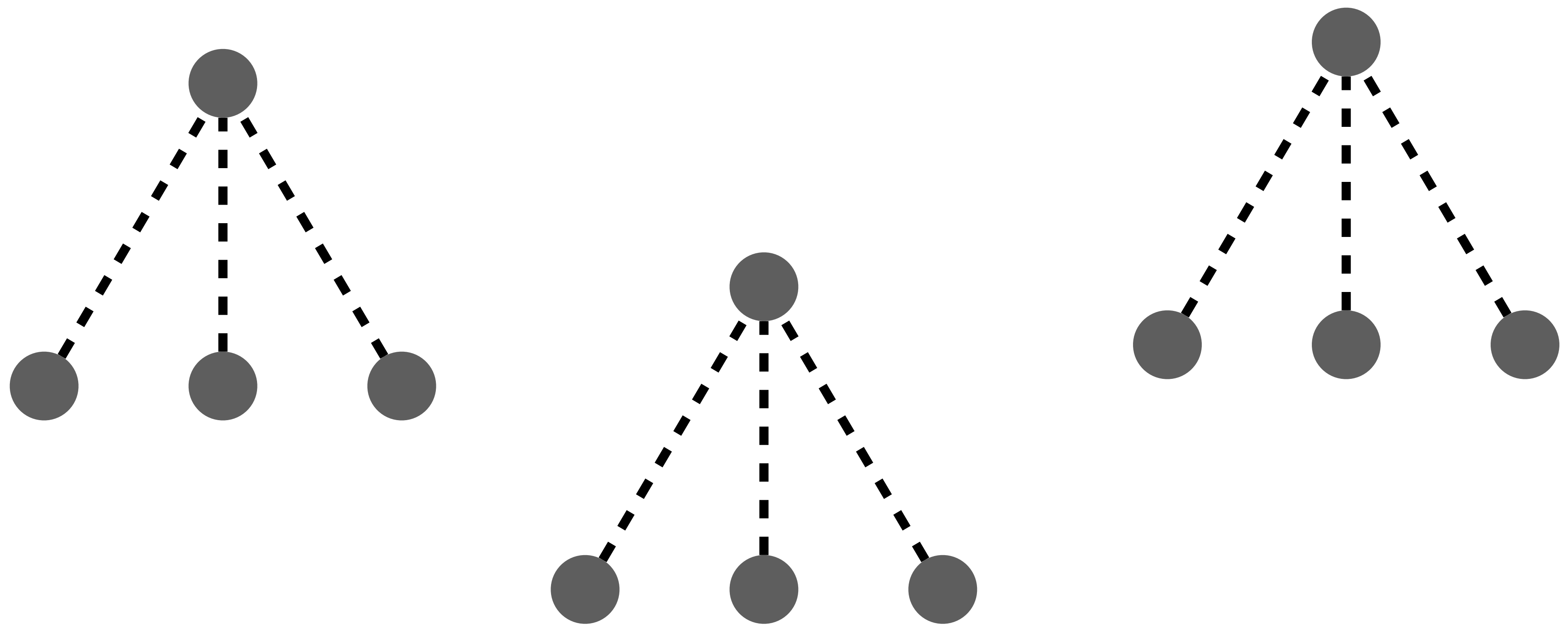
Main observation:

It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges



uncolored edges
scattered around



uncolored edges
forming **stars**

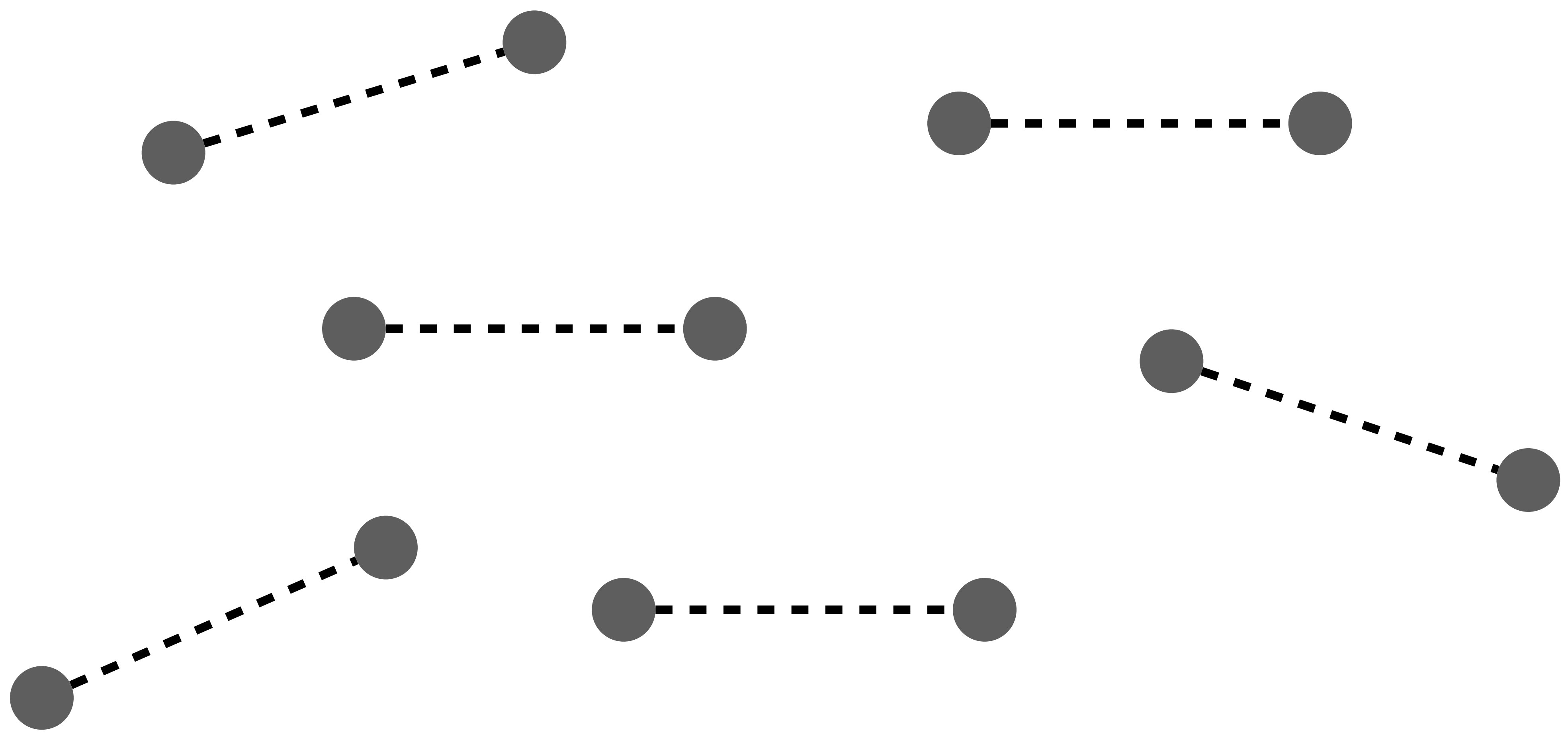
New Approach: Coloring Stars

<u>Main observation:</u>

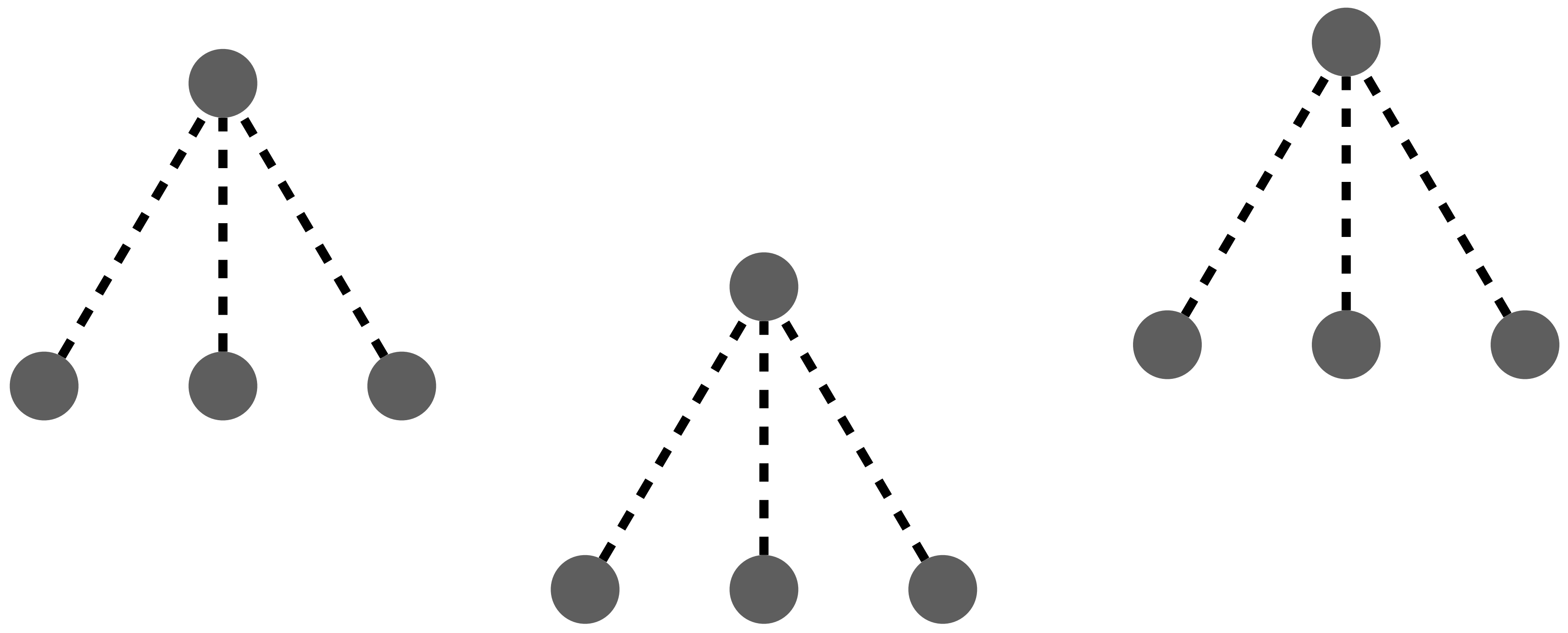
It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges

There are ways to reach this **precondition** efficiently



uncolored edges scattered around



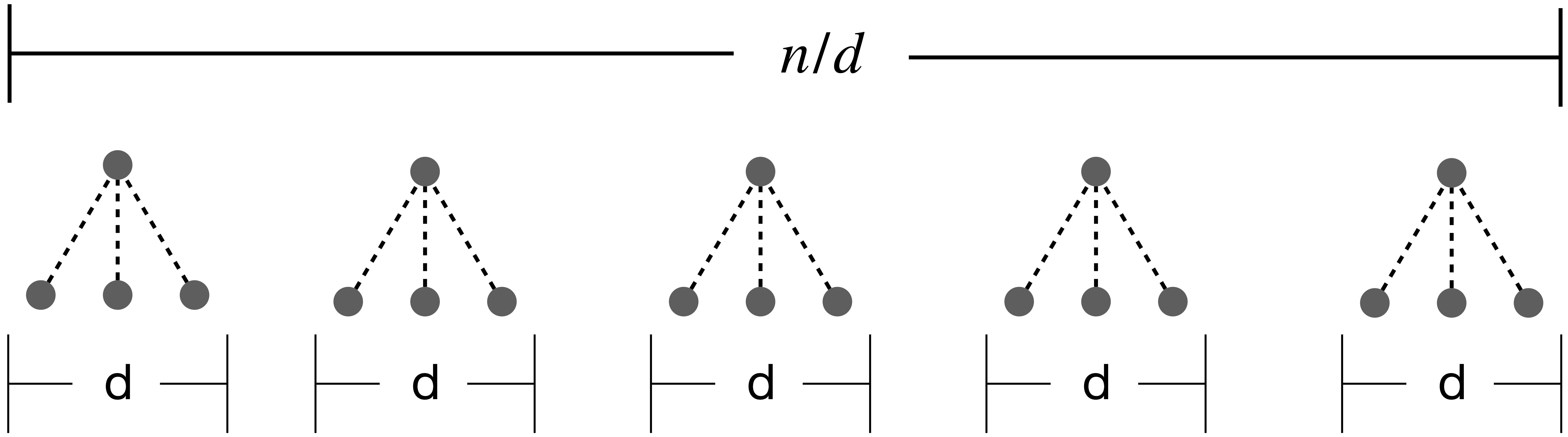
uncolored edges forming **stars**

New Approach: Coloring Stars

Main observation:

It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges (**bipartite** graphs)

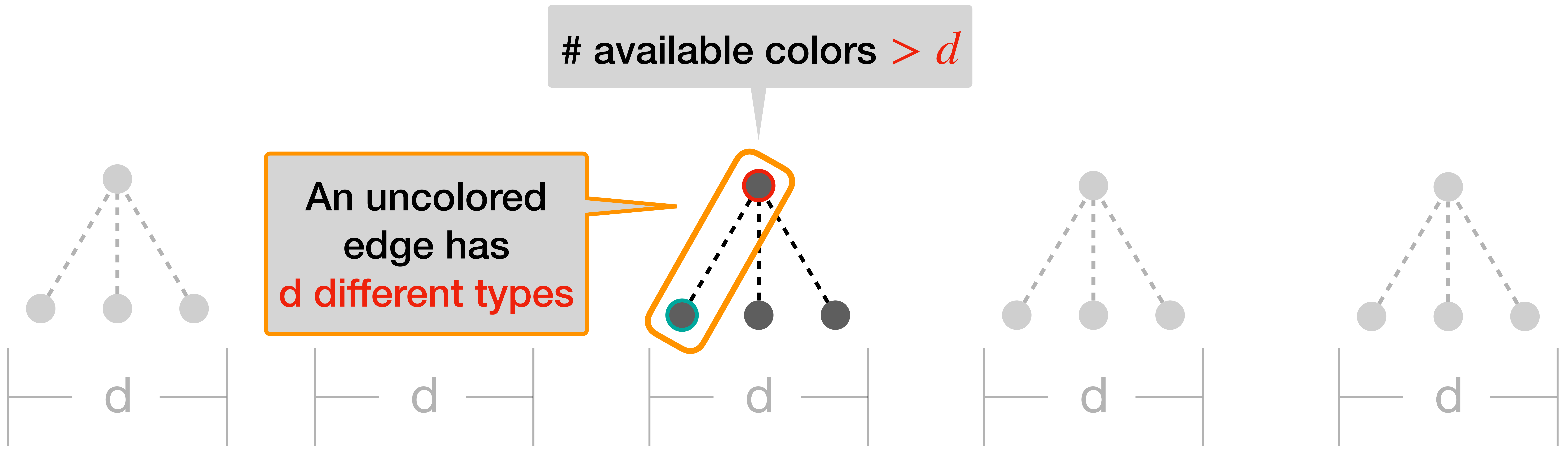


New Approach: Coloring Stars

<u>**Main observation:**</u>

It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges (**bipartite** graphs)

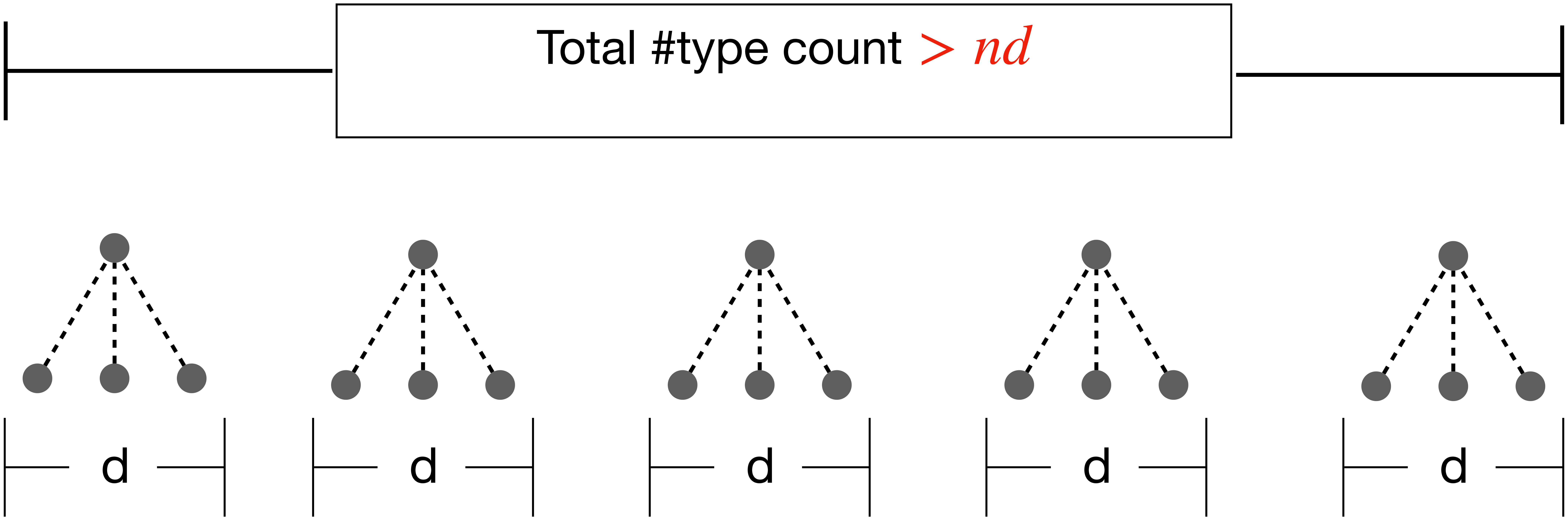


New Approach: Coloring Stars

Main observation:

It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges (**bipartite** graphs)



New Approach: Coloring Stars

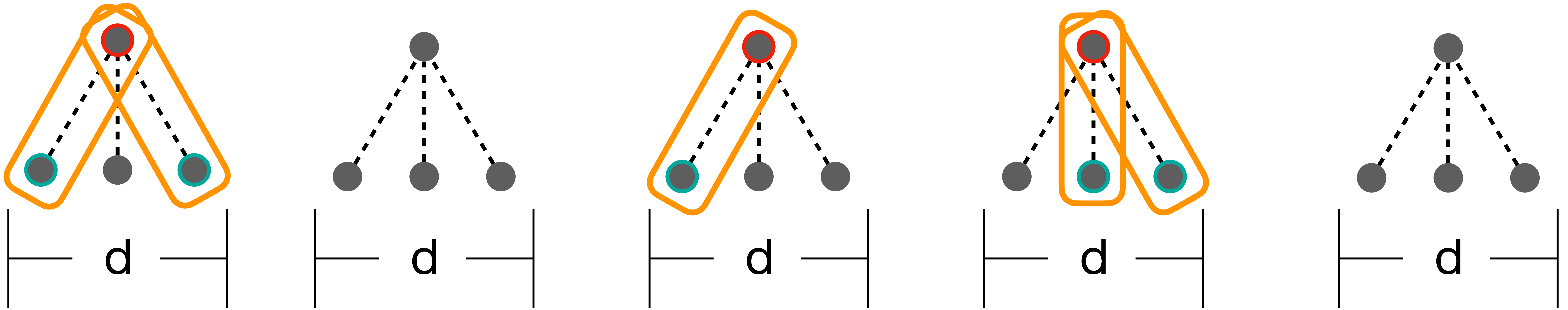
Main observation:

It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges (**bipartite** graphs)

Total #type count $> nd$

An **average** type has $\geq nd/\Delta^2$ edges



New Approach: Coloring Stars

Main observation:

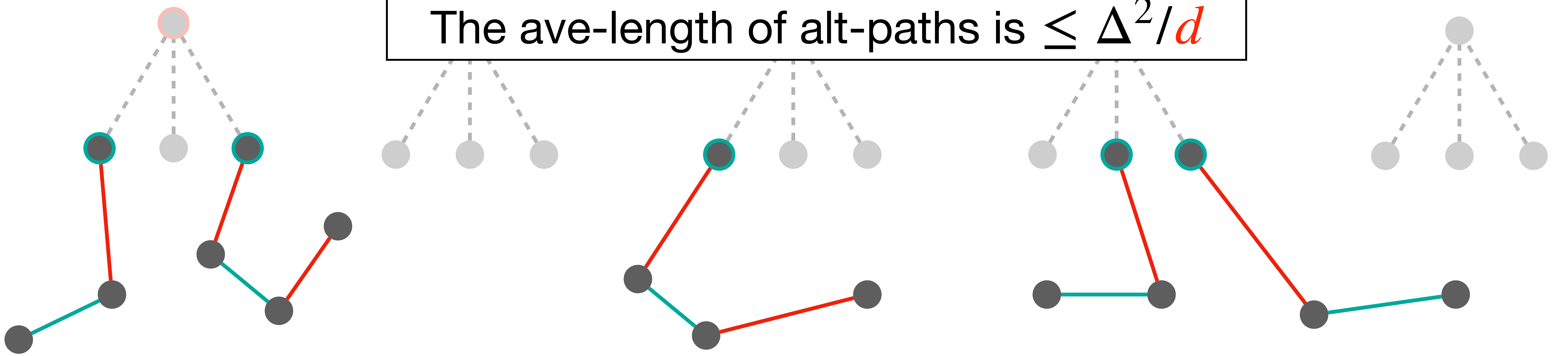
It could be **easier** if uncolored edges form **star subgraphs**

Reason: An average type contains **more** uncolored edges (**bipartite** graphs)

Total #type count $> nd$

An **average** type has $\geq nd/\Delta^2$ edges

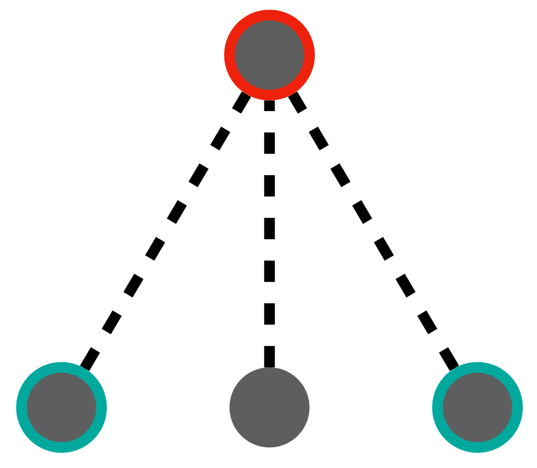
The ave-length of alt-paths is $\leq \Delta^2/d$

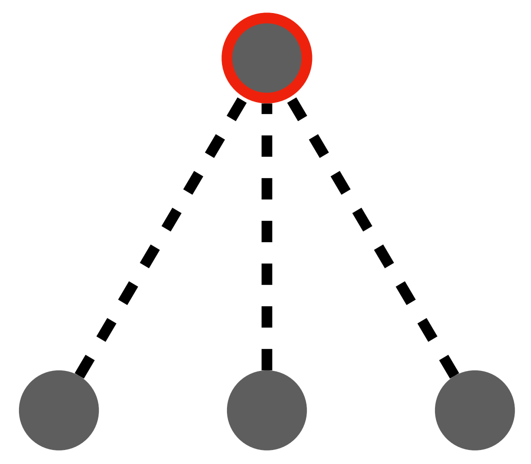


Coloring Stars in General Graphs

Issue in general graphs:

- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**



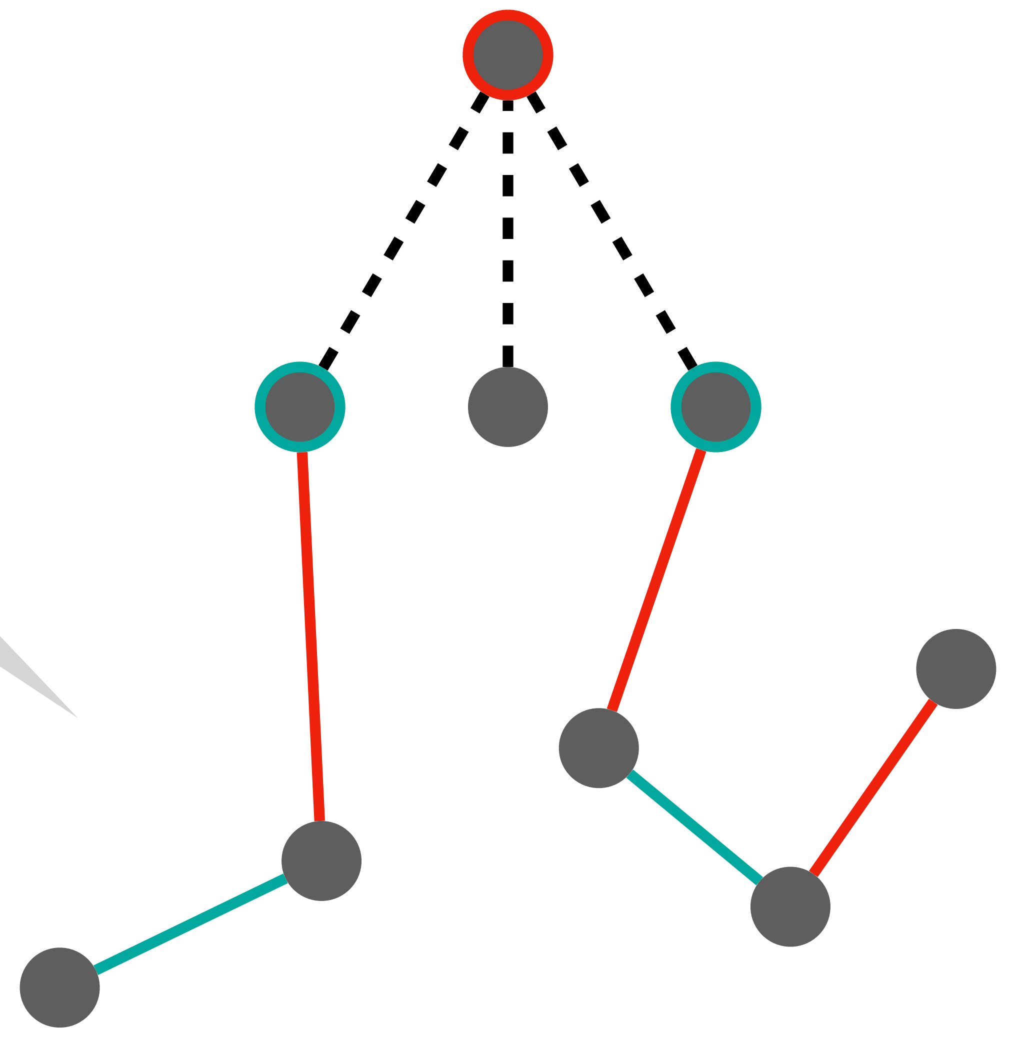


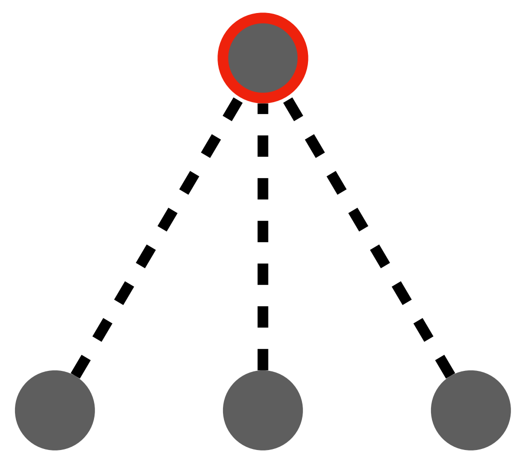
Coloring Stars in General Graphs

Issue in general graphs:

- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**

In **bipartite** graphs, edges always generate **different alternating paths**



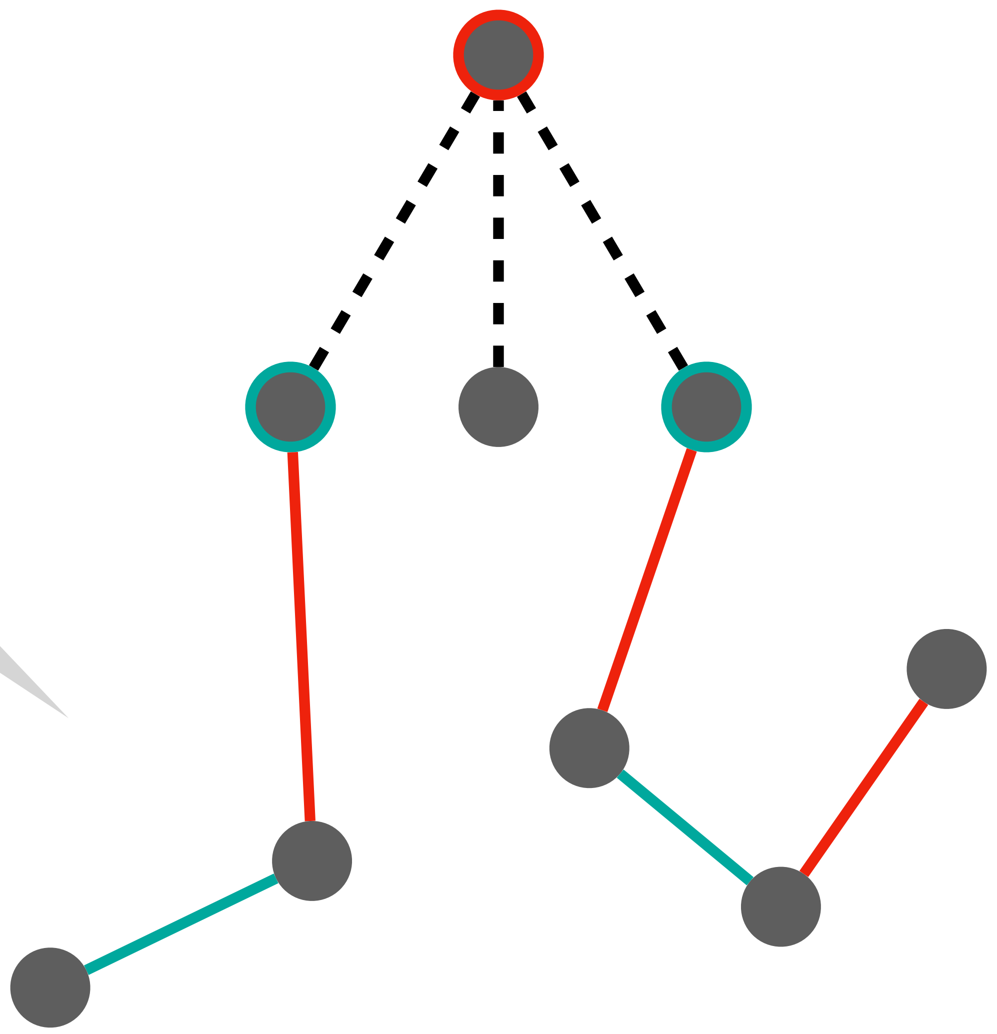


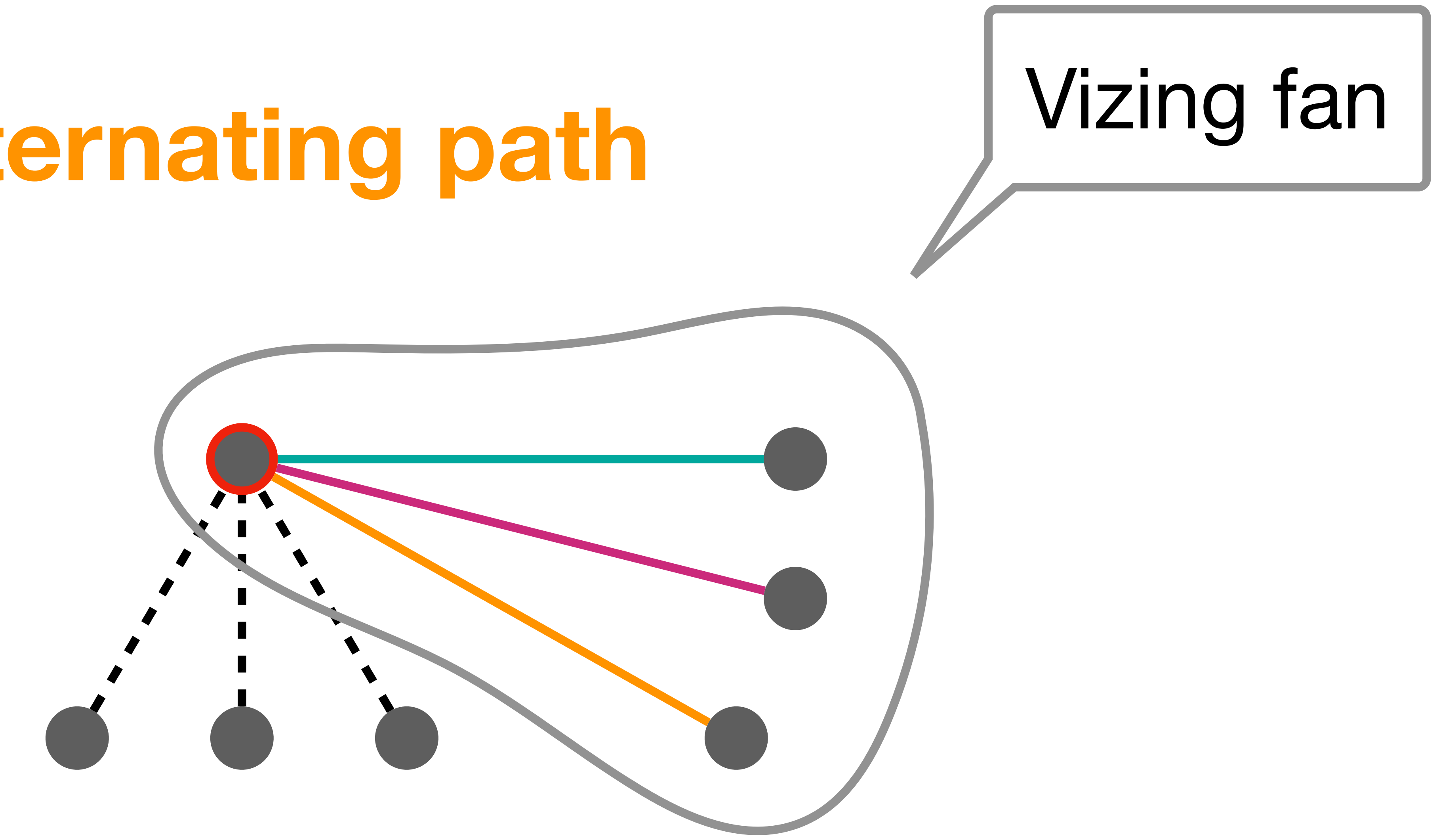
Coloring Stars in General Graphs

Issue in general graphs:

- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**

In **bipartite** graphs, edges always generate **different alternating paths**



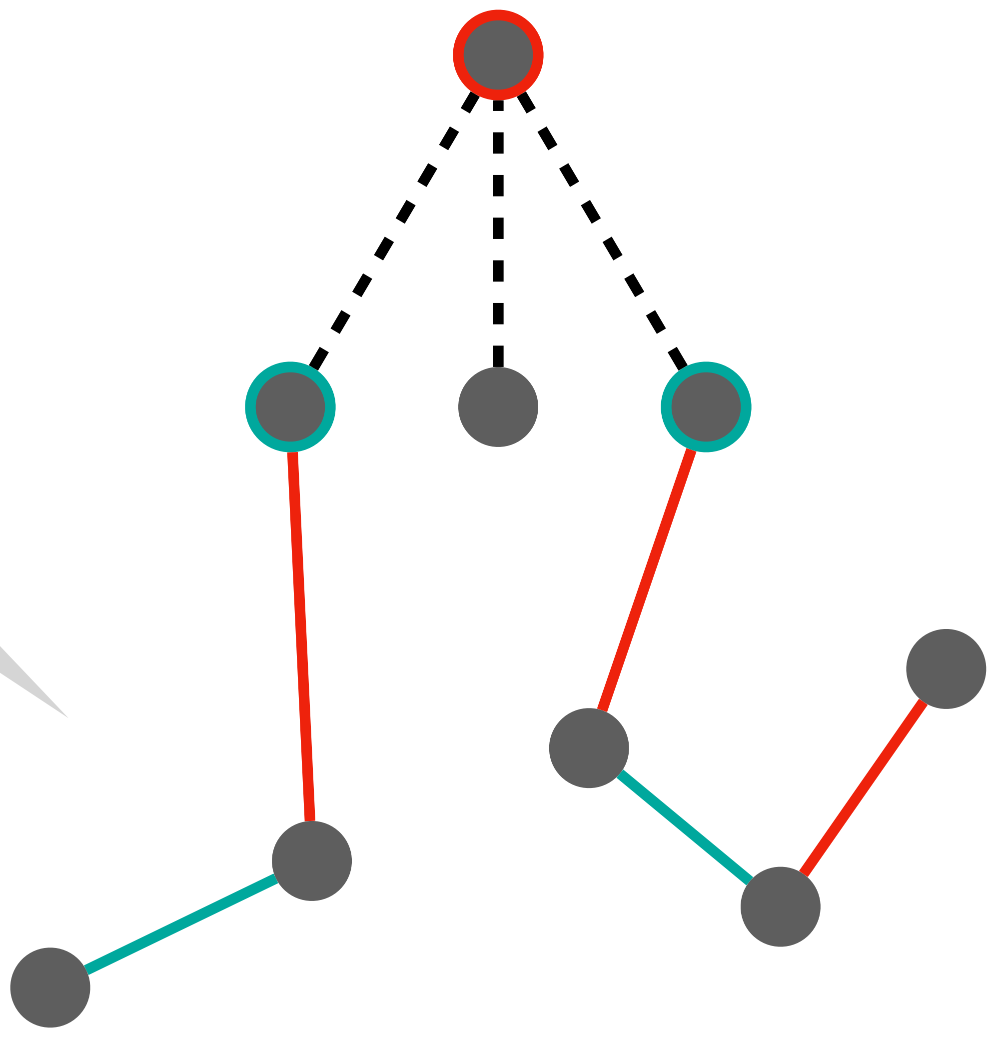


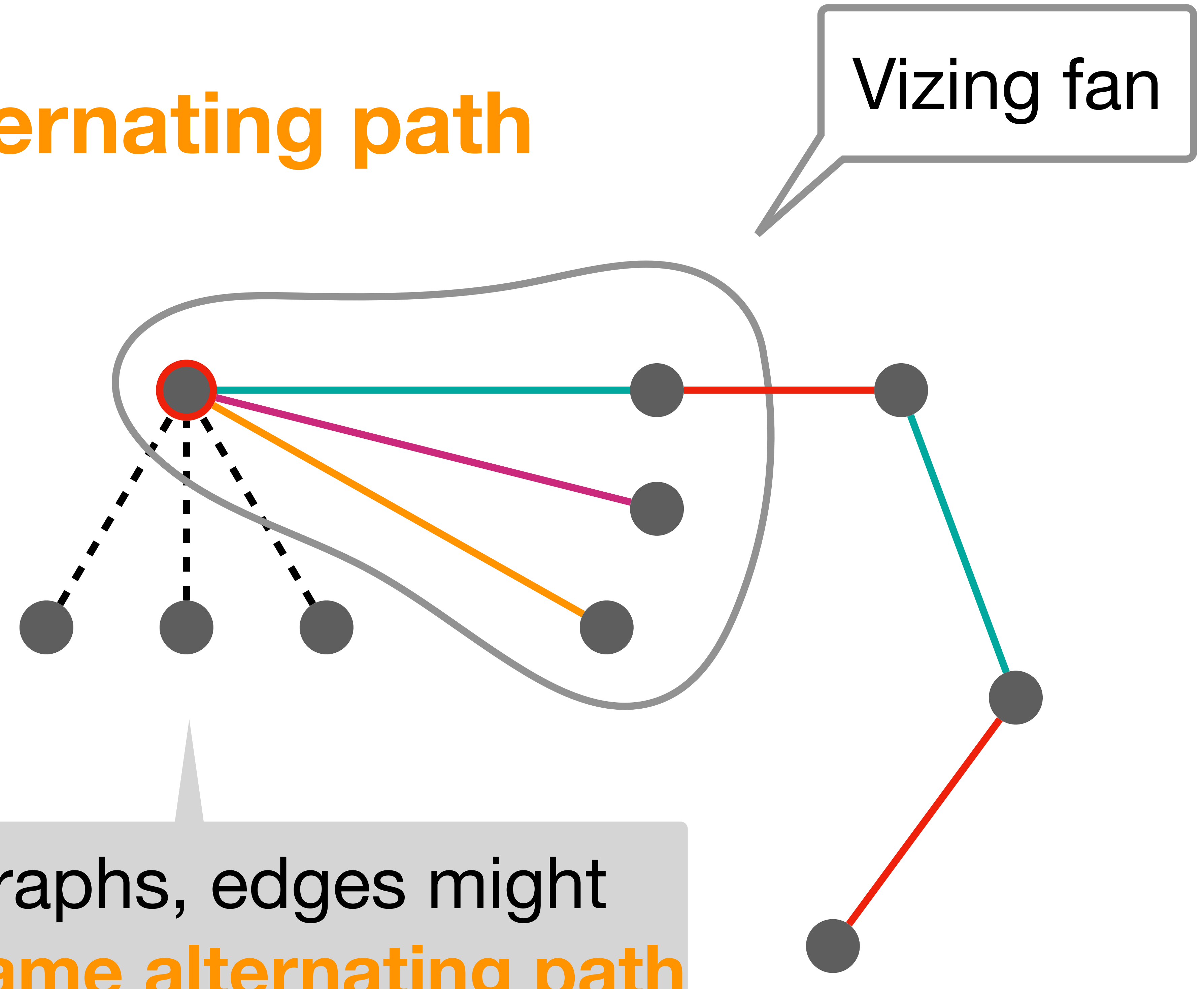
Coloring Stars in General Graphs

Issue in general graphs:

- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**

In **bipartite** graphs, edges always generate **different alternating paths**



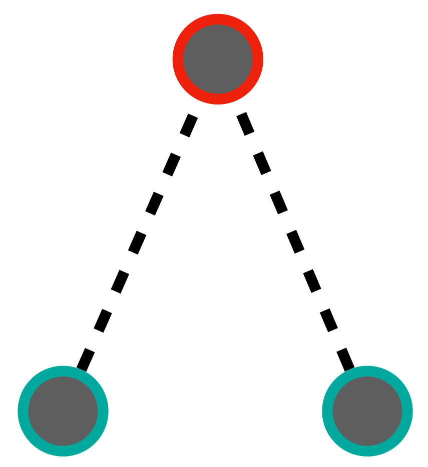


In **general** graphs, edges might generate the **same alternating path**

Coloring Stars in General Graphs

Observation:

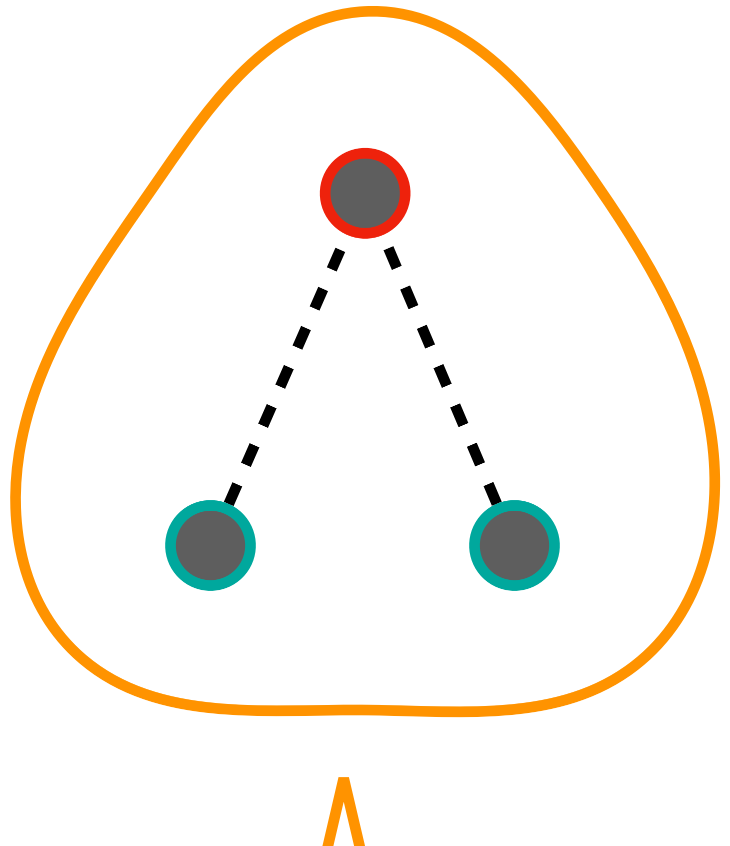
- Use **u-fans** to avoid Vizing fans and go back to bipartite case [Gabow et al., 1985]



Coloring Stars in General Graphs

Observation:

- Use **u-fans** to avoid Vizing fans and go back to bipartite case [Gabow et al., 1985]

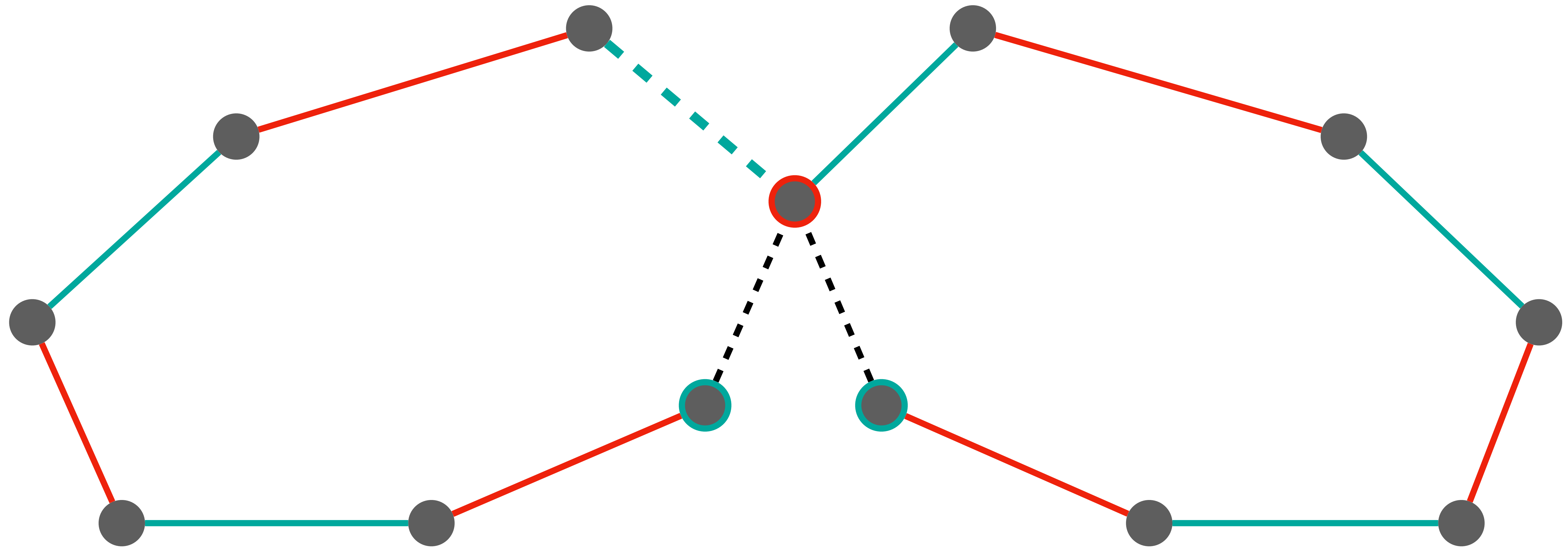


u-fan: a pair of leaves
missing the same color

Coloring Stars in General Graphs

Observation:

- Use **u-fans** to avoid Vizing fans and go back to bipartite case [Gabow et al., 1985]

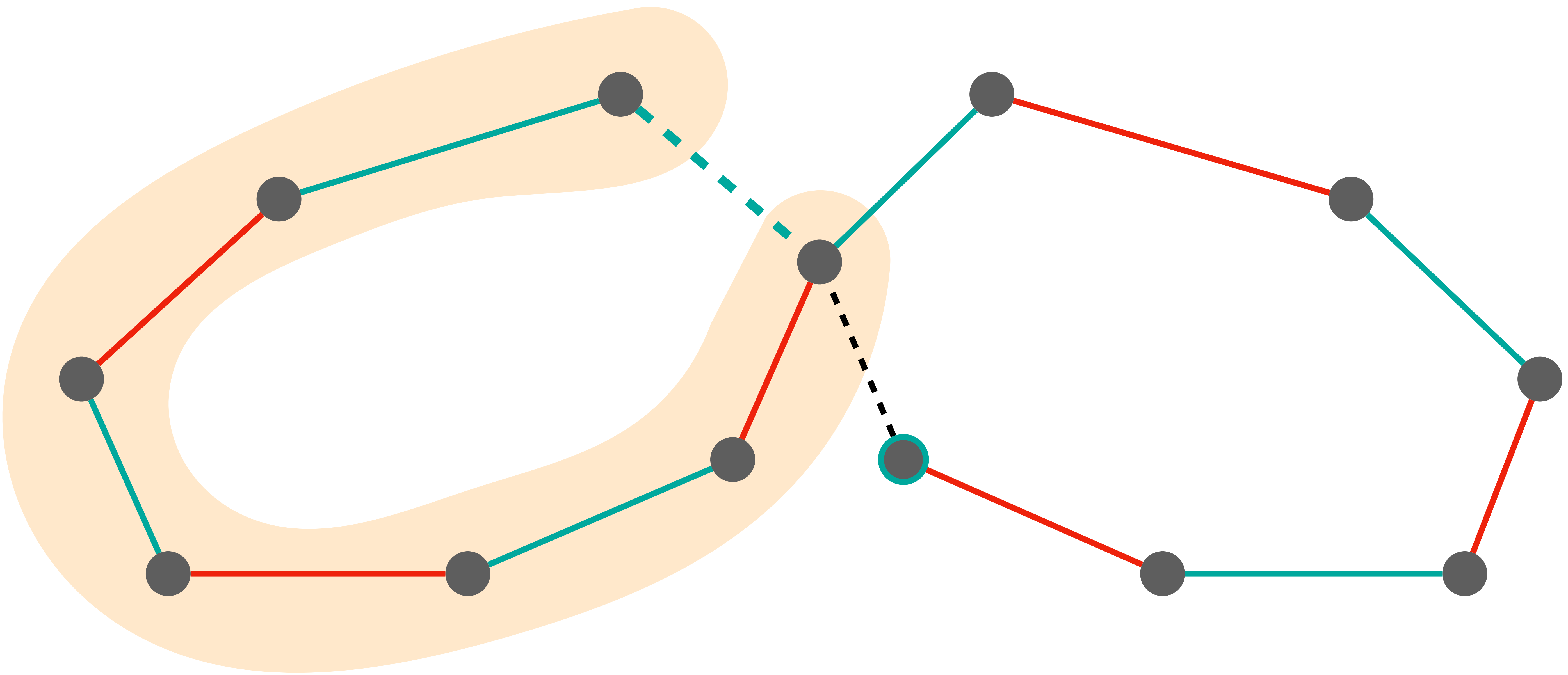


One of the alternating path must
be useful for color extension

Coloring Stars in General Graphs

Observation:

- Use u-fans to avoid Vizing fans and go back to bipartite case [Gabow et al., 1985]



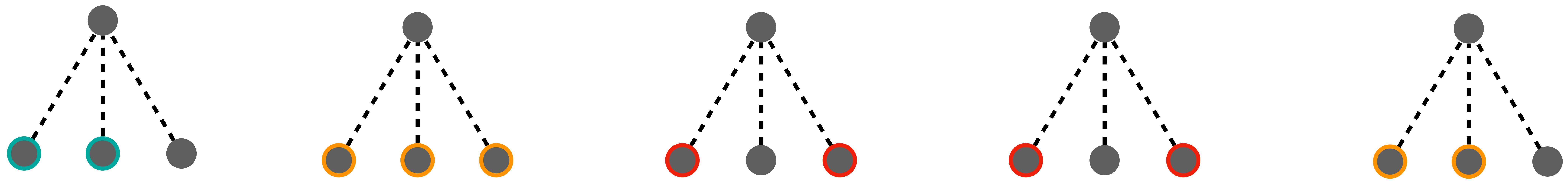
Flip this path and extend the color

One of the alternating path must be useful for color extension

Coloring Stars in General Graphs

Issue in general graphs:

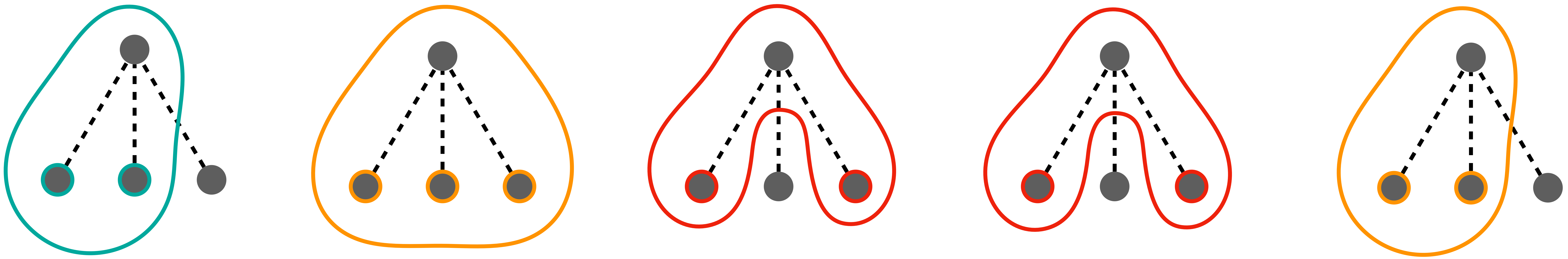
- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**



Coloring Stars in General Graphs

Issue in general graphs:

- Edge types are defined by the **Vizing fan** structure
- Edges in the star might generate the **same alternating path**

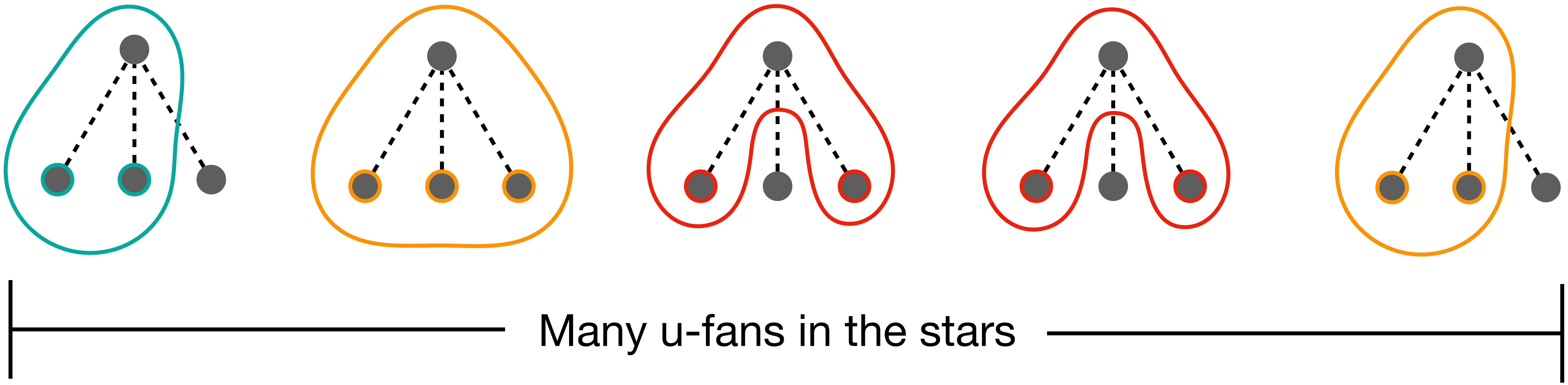


Many u-fans in the stars

Coloring Stars in General Graphs

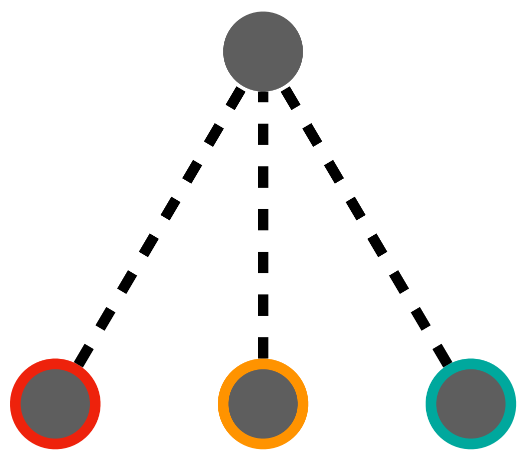
<u>Issue in general graphs:</u>

- ~~Edge types are defined by the **Vizing fan** structure~~ Use u-fans instead
- ~~Edges in the star might generate the **same alternating path**~~ Bipartite analysis

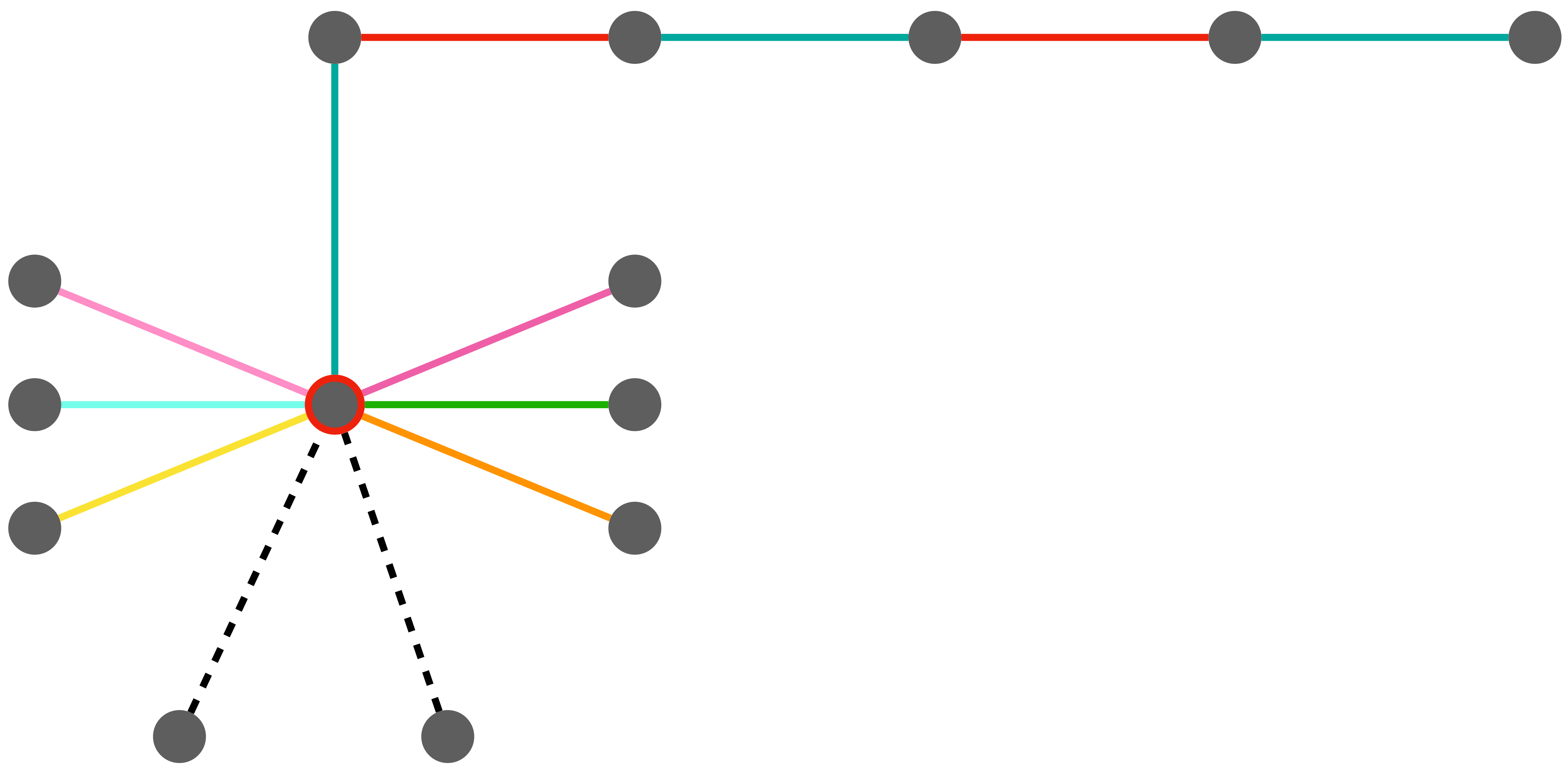


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

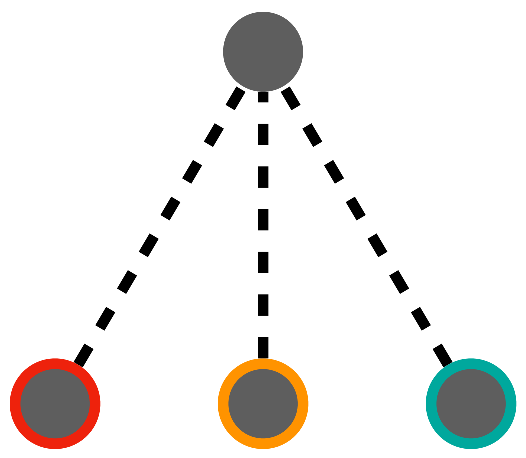


Suppose two uncolored edges share a Vizing chain

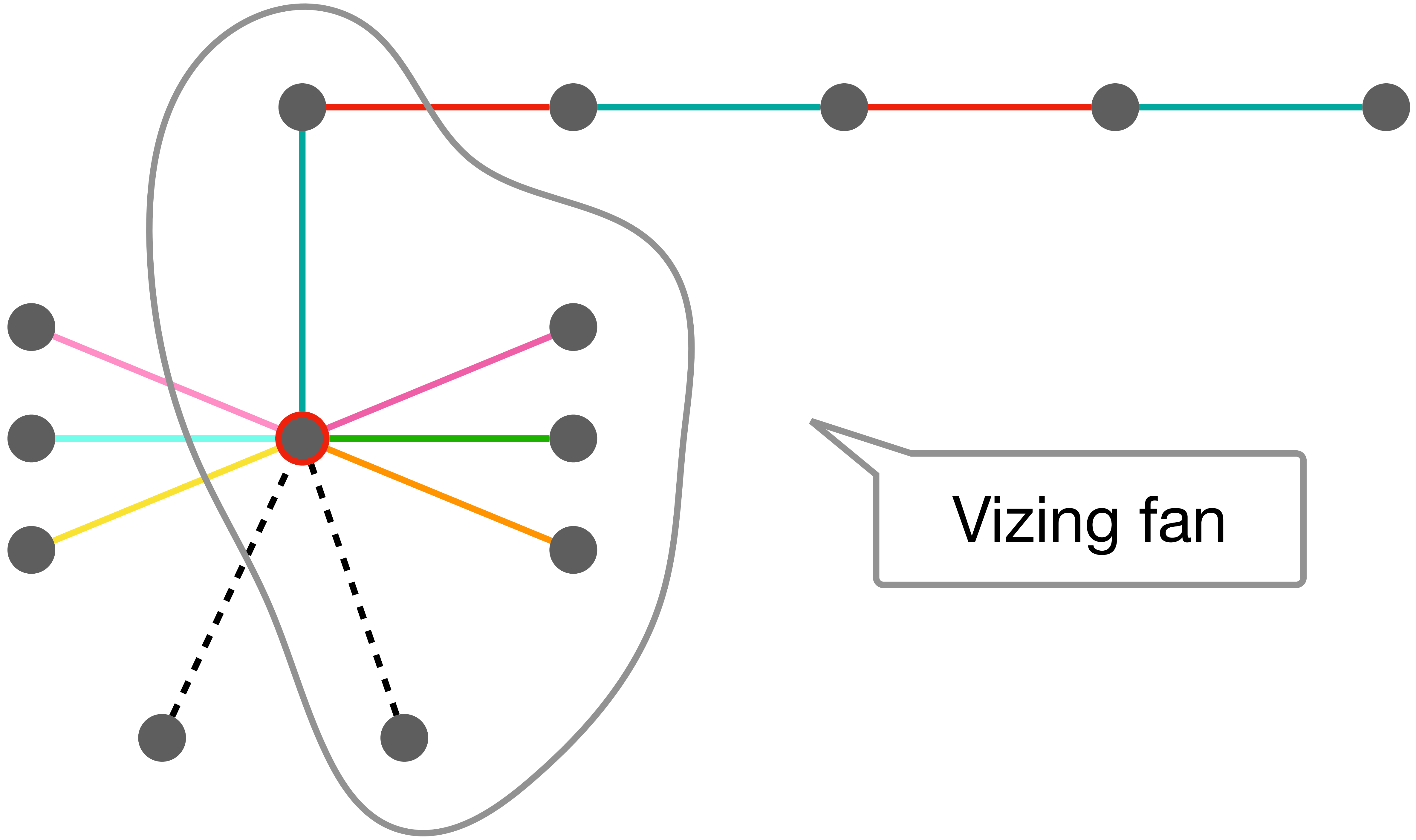


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

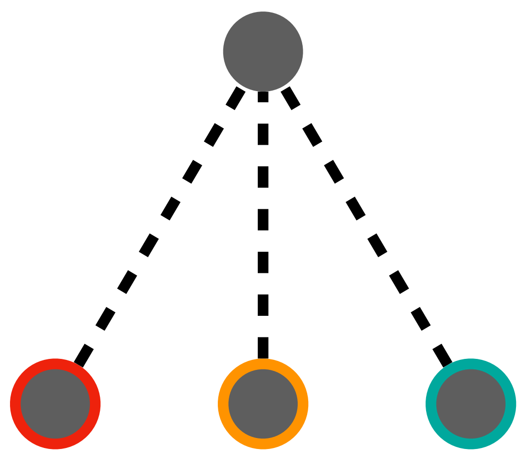


Suppose two uncolored edges share a Vizing chain

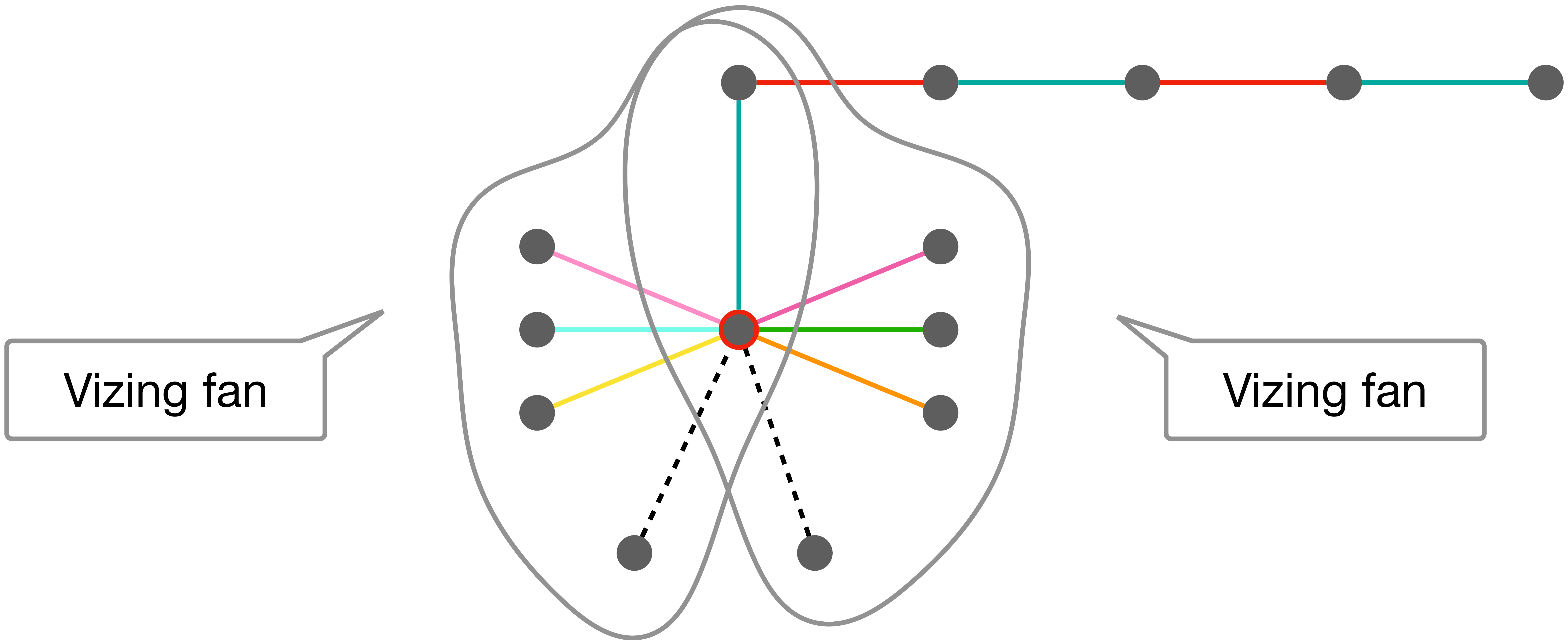


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

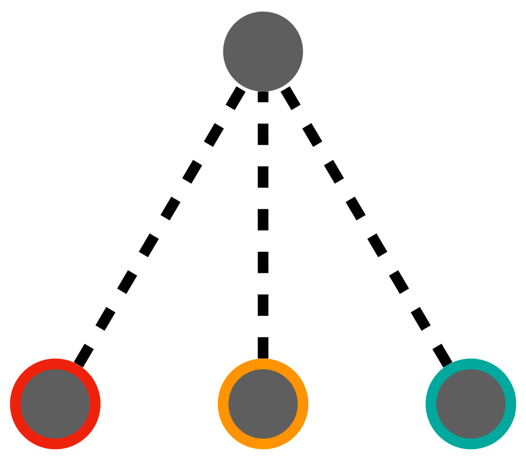


Suppose two uncolored edges share a Vizing chain

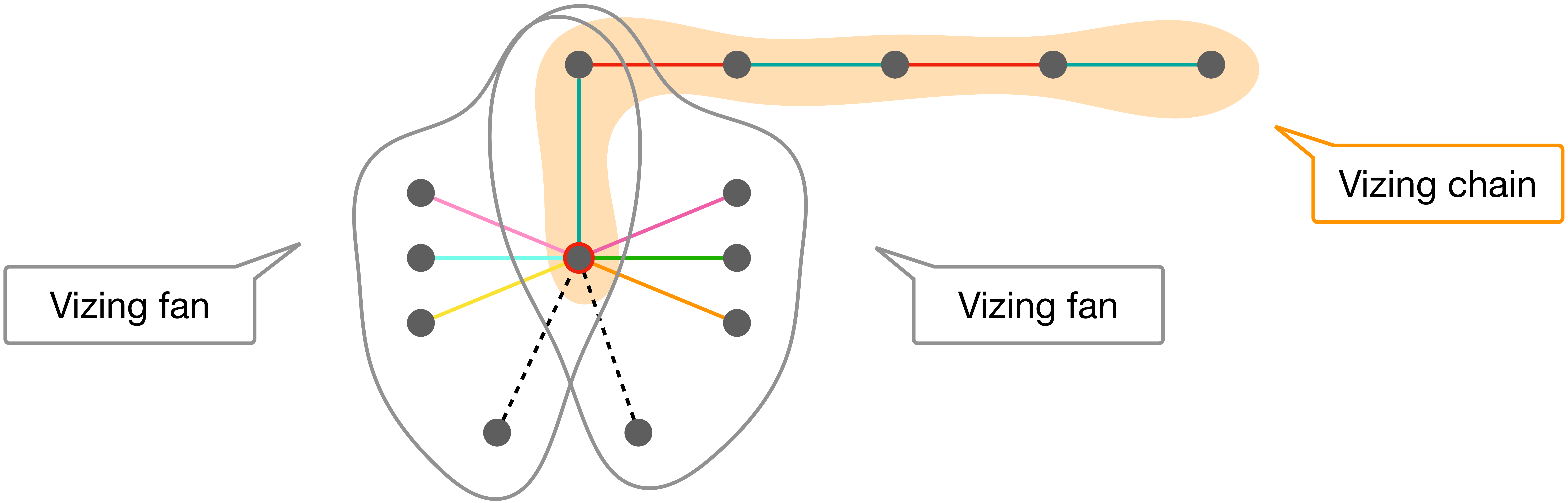


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

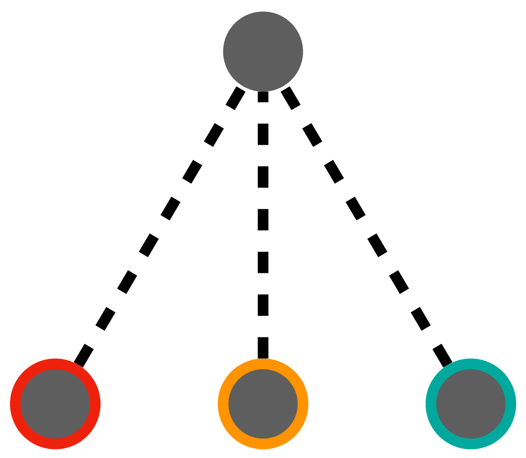


Suppose two uncolored edges share a Vizing chain

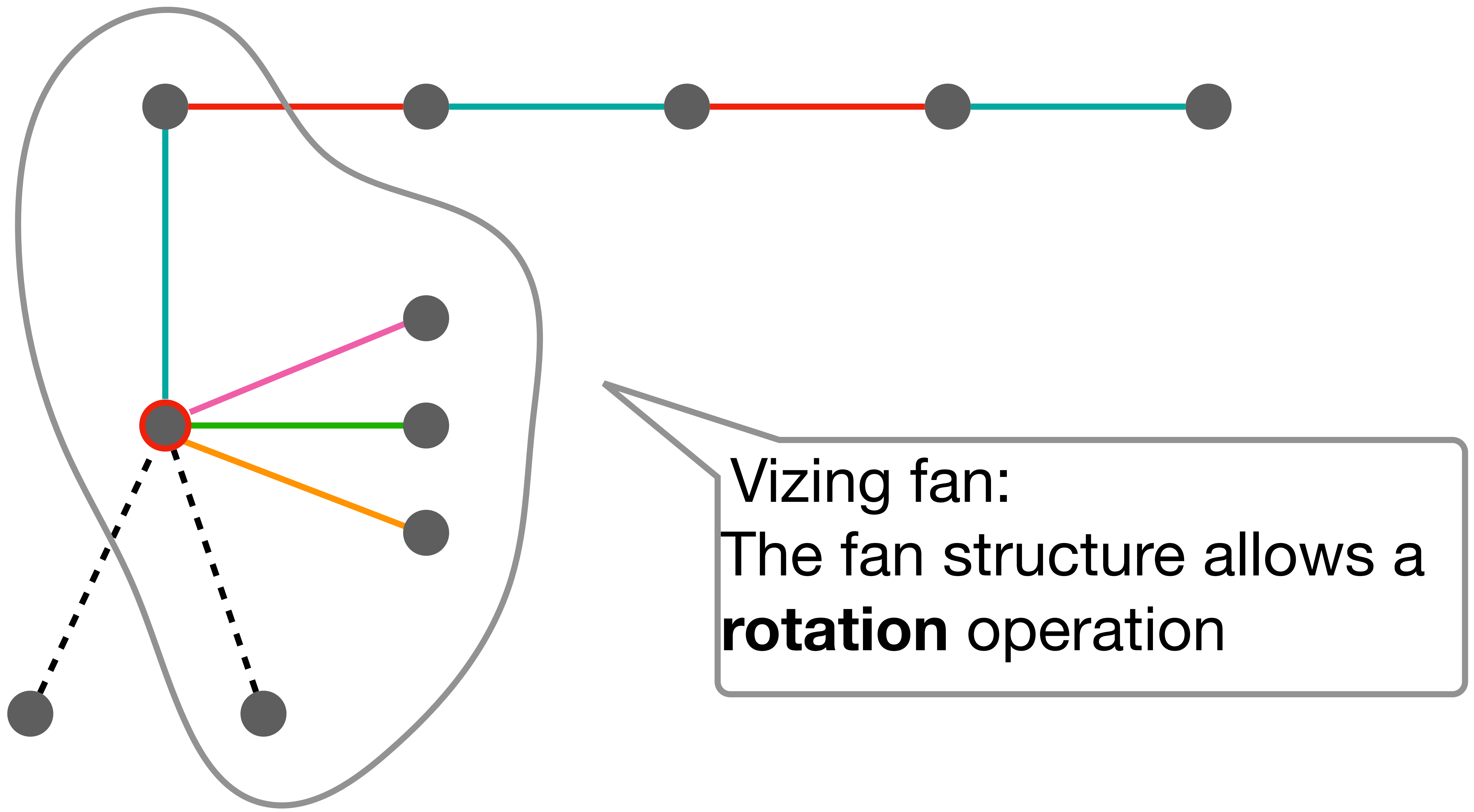


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

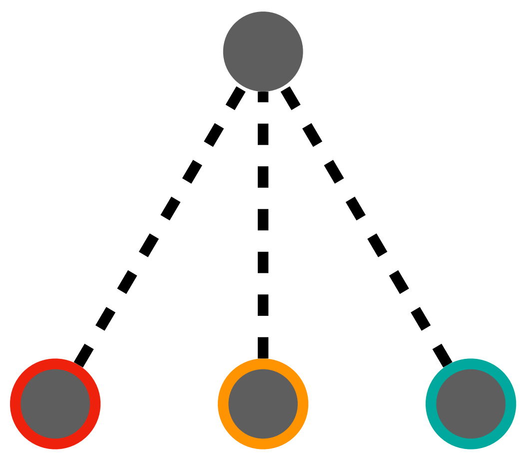


Suppose two uncolored edges share a Vizing chain

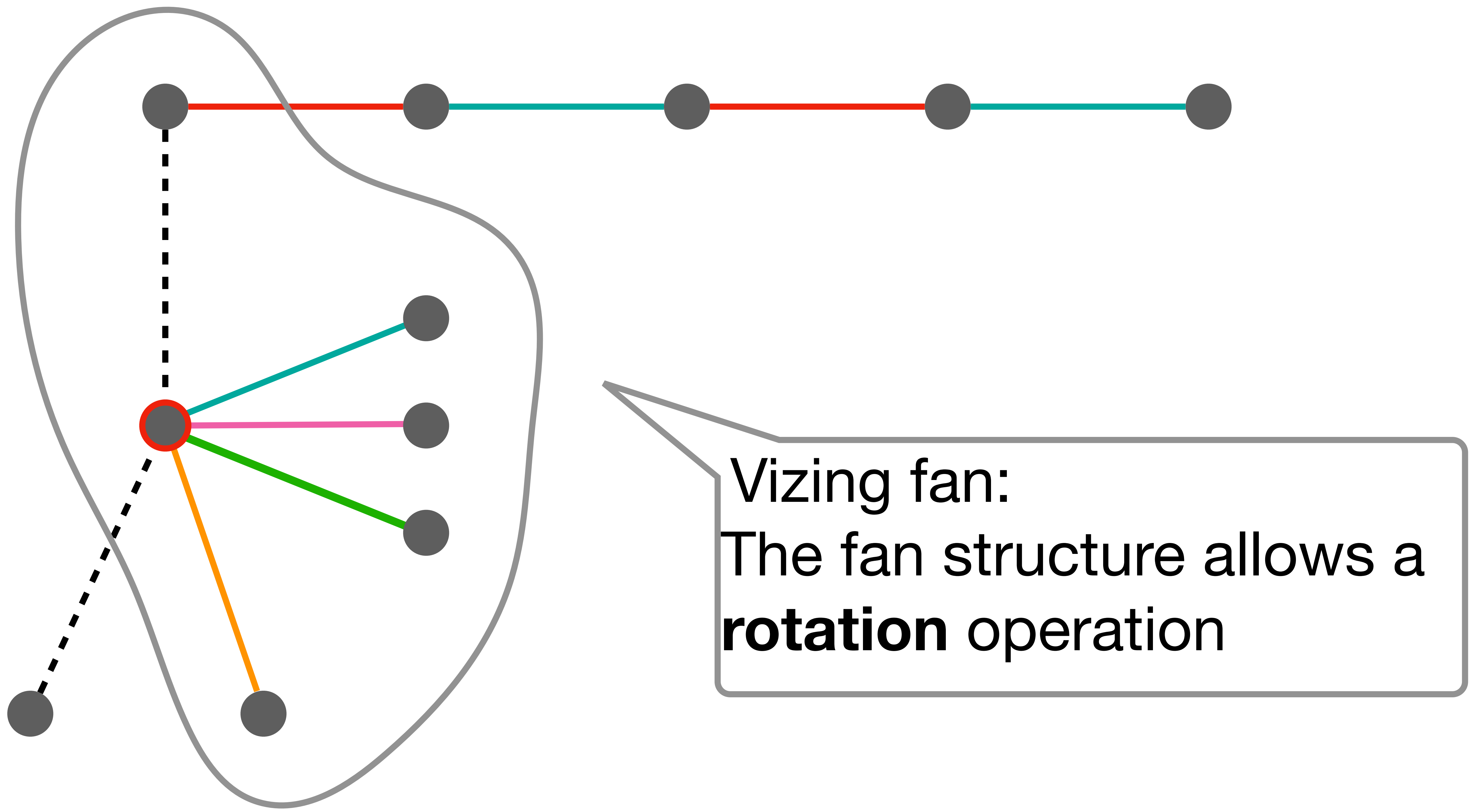


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?

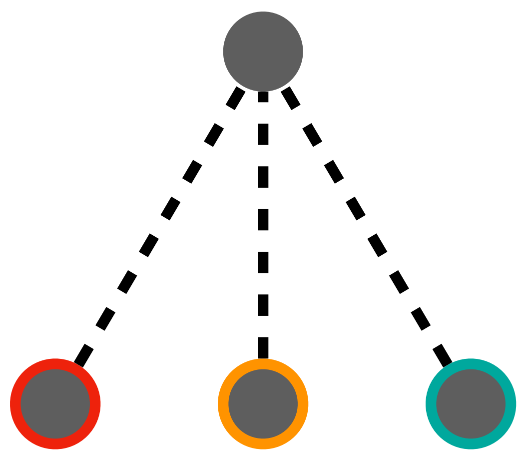


Suppose two uncolored edges share a Vizing chain

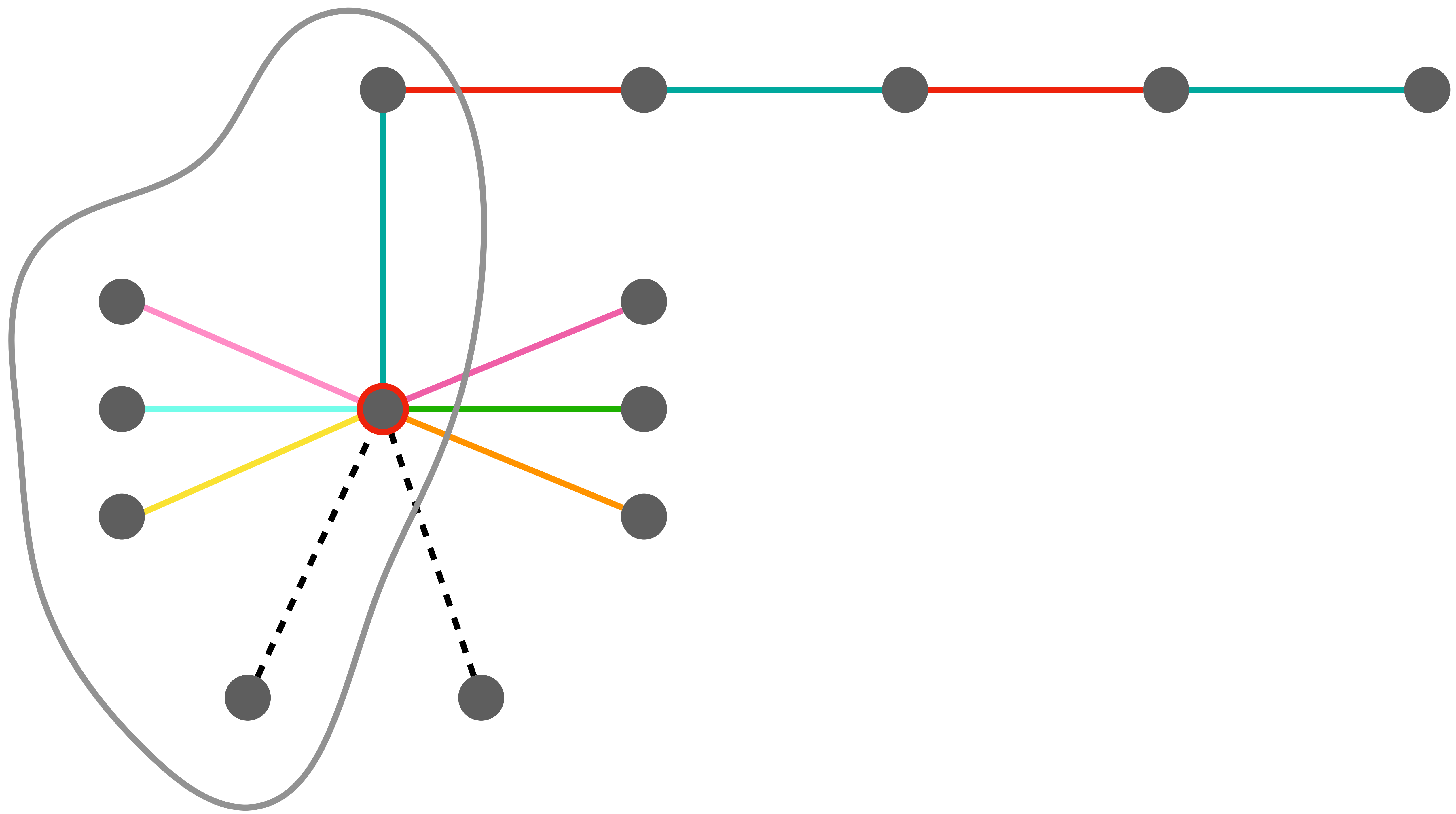


Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?



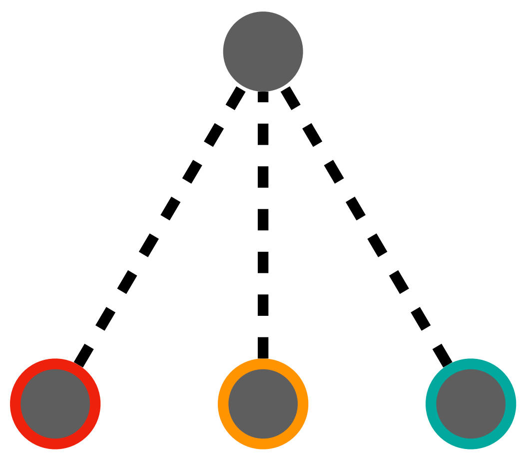
Suppose two uncolored edges share a Vizing chain



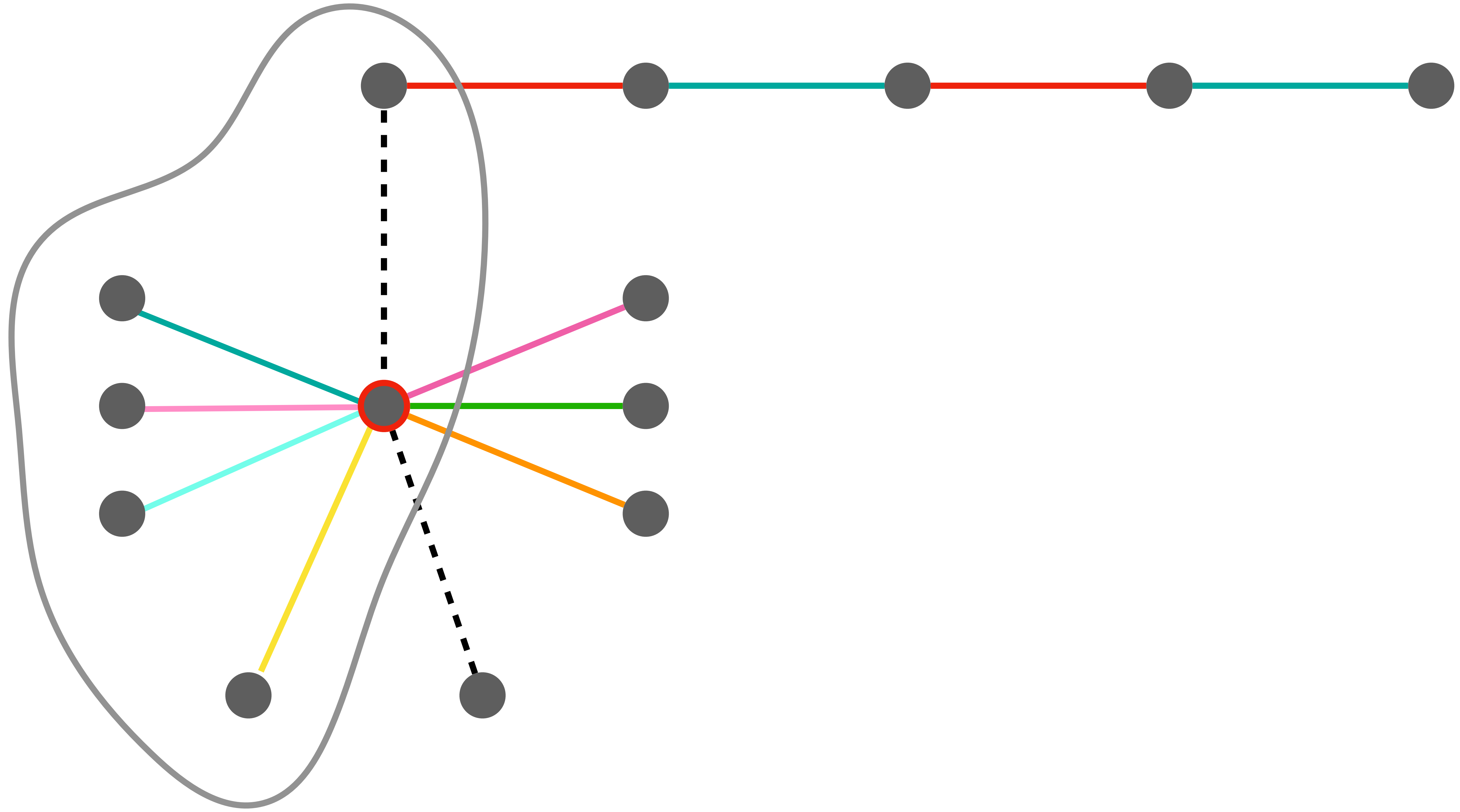
Vizing fan:
The fan structure allows a **rotation** operation

Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?



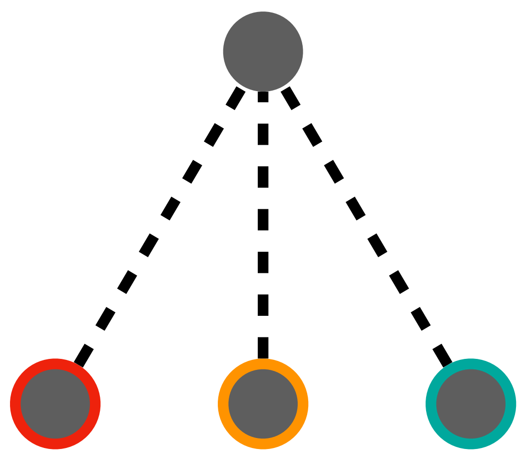
Suppose two uncolored edges share a Vizing chain



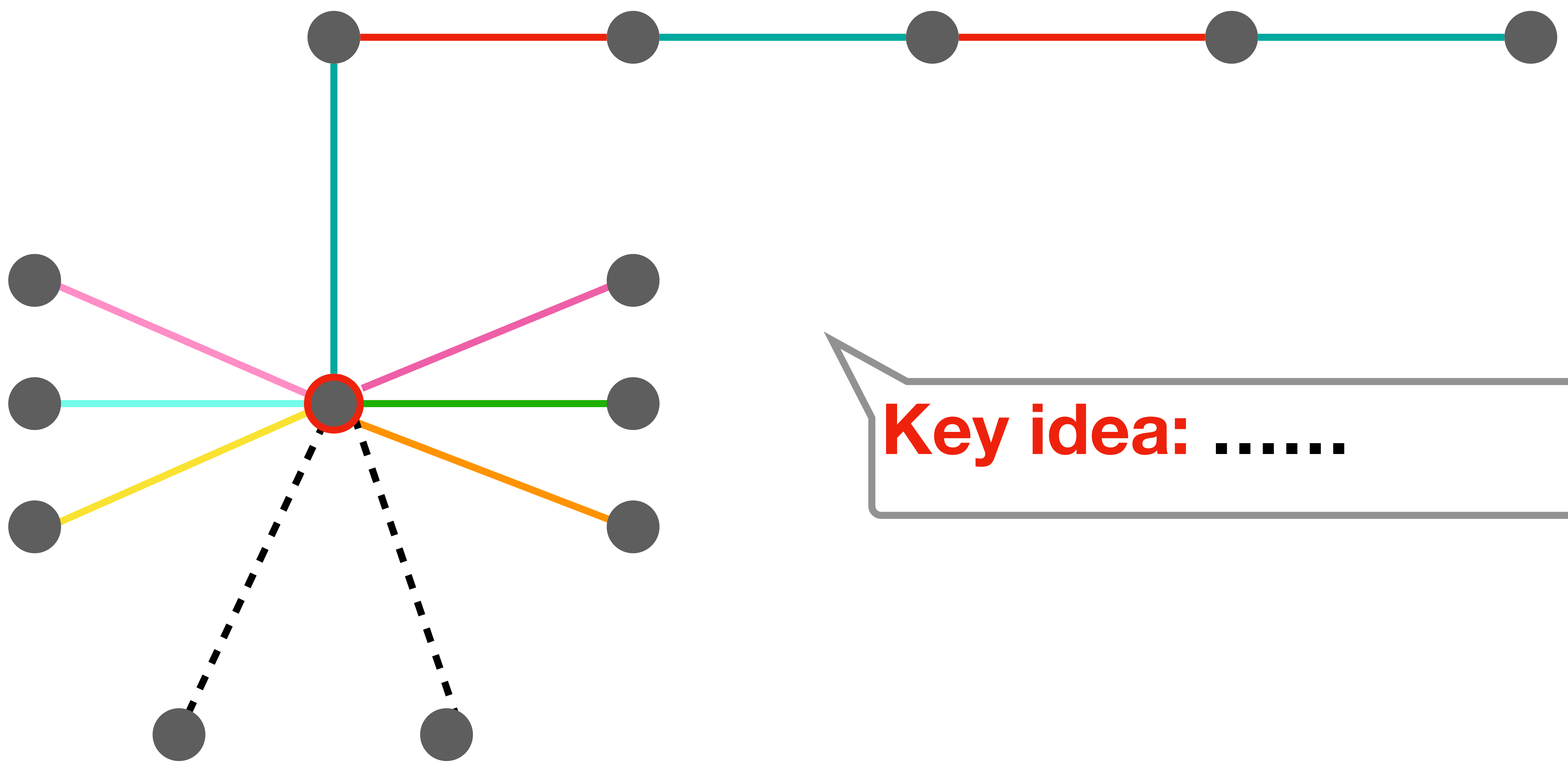
Vizing fan:
The fan structure allows a **rotation** operation

Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?



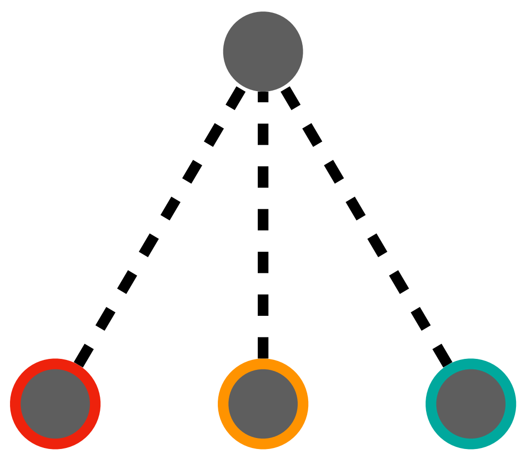
Suppose two uncolored edges share a Vizing chain



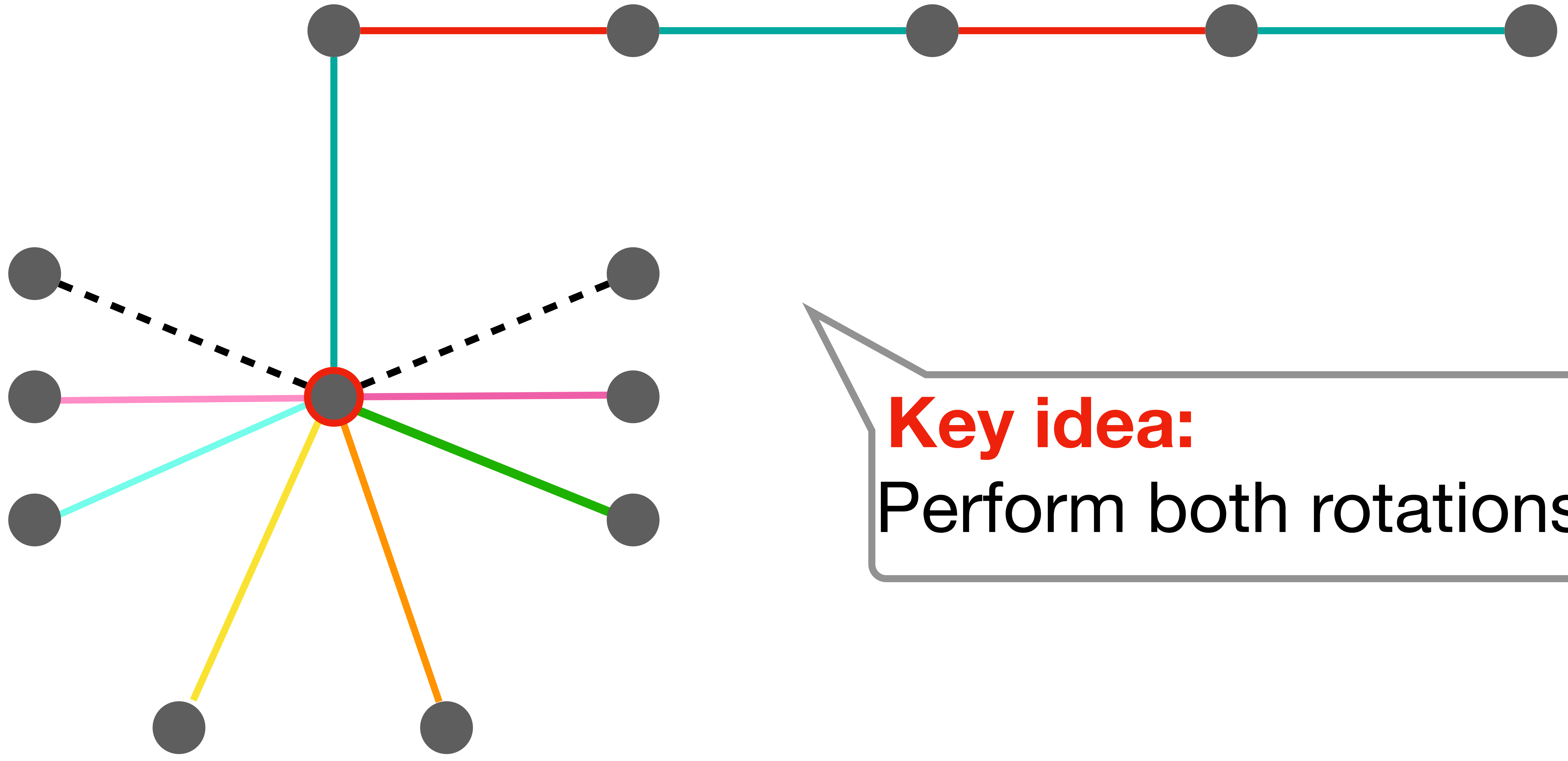
Key idea: ……

Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?



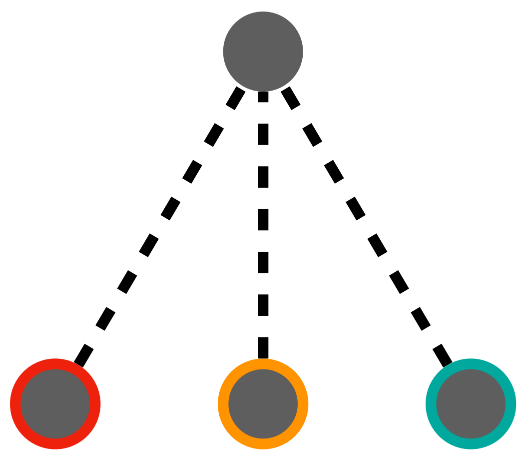
Suppose two uncolored edges share a Vizing chain



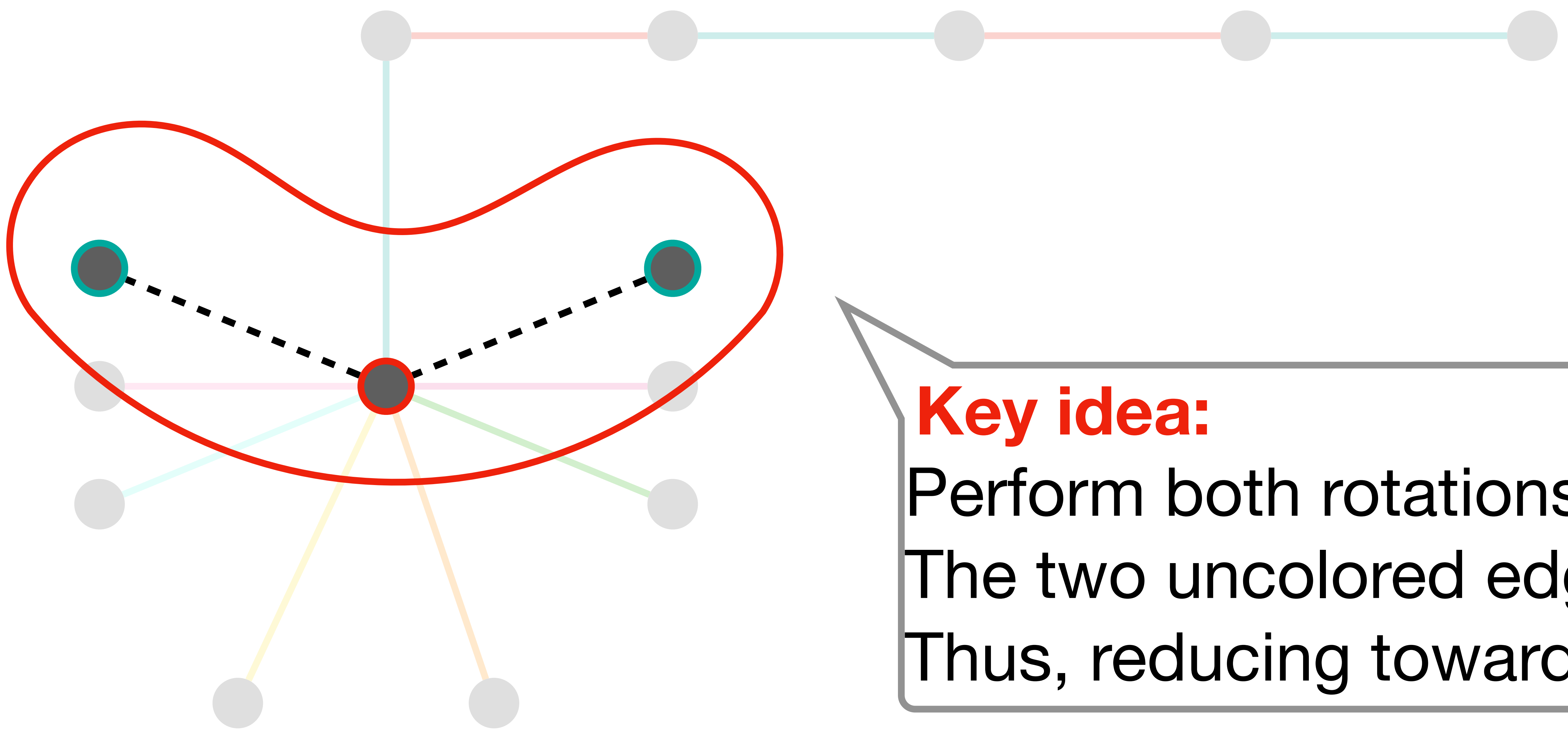
Key idea:
Perform both rotations except for the last edge

Coloring Stars in General Graphs

What if leaf **missing colors are different** in the same star?



Suppose two uncolored edges share a Vizing chain



Key idea:
Perform both rotations except for the last edge
The two uncolored edges make a new **u-fan**
Thus, reducing towards the **bipartite** case

Conclusion

- Main result: $(\Delta + 1)$-edge coloring faster than the classical $mn^{1/2}$ bound
- Open question: Near-linear runtime?
- Other questions: Dynamic? Parallel?

Conclusion

- Main result: $(\Delta + 1)$-edge coloring faster than the classical $mn^{1/2}$ bound
- ~~Open question: Near-linear runtime?~~ Solved by more recent [ABBCSZ’24]
- Other questions: Dynamic? Parallel?