On connections between k-coloring and Euclidean k-means

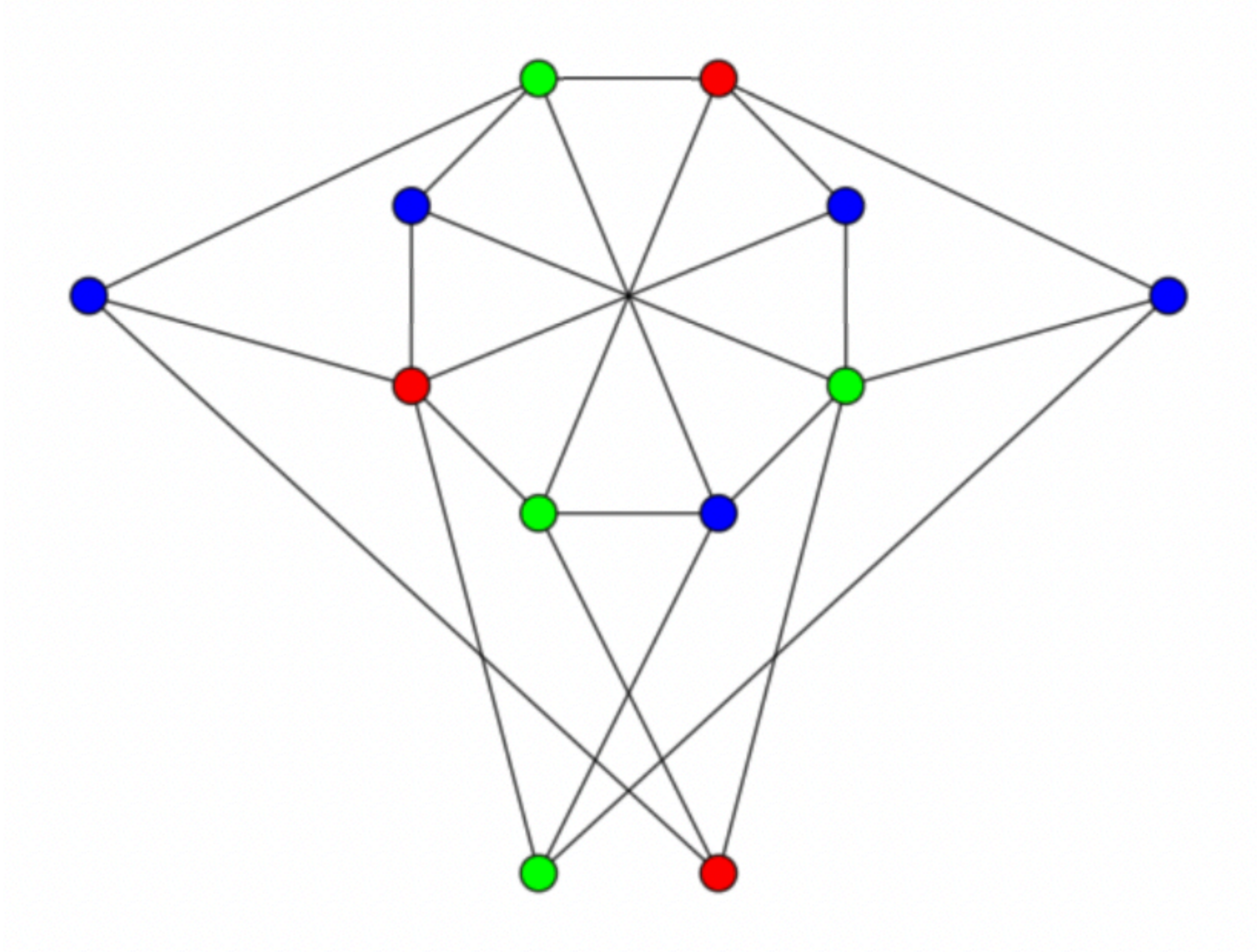
Karthik C. S.
(Rutgers University)

Joint Work with

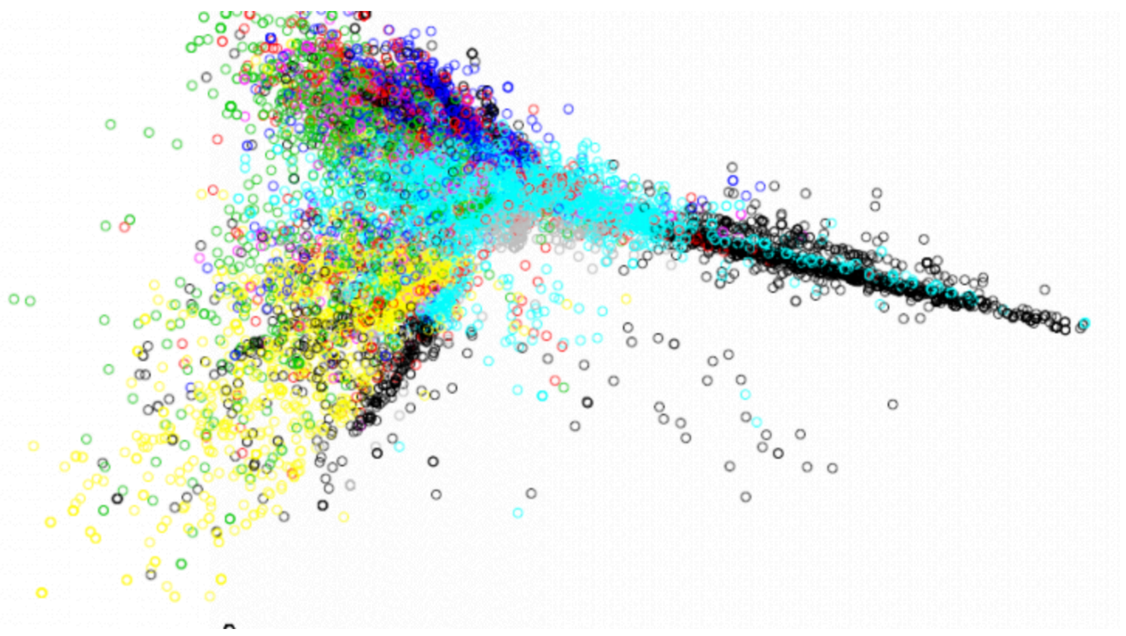
Enver Aman
Undergrad at Rutgers
Graduated May 2024

Sharath Punna
Masters at Rutgers
Graduated May 2023

Graph Coloring

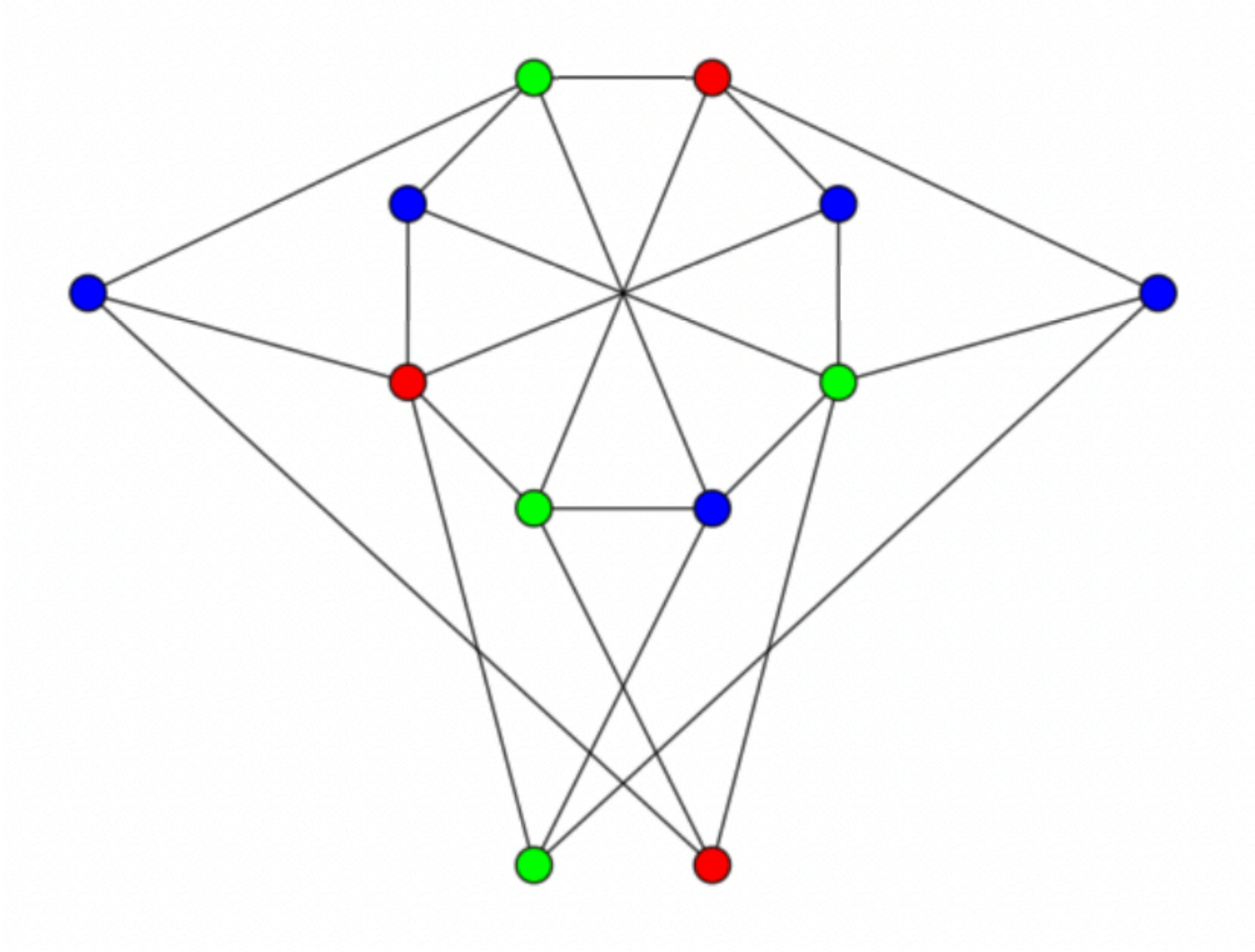


k-means Clustering



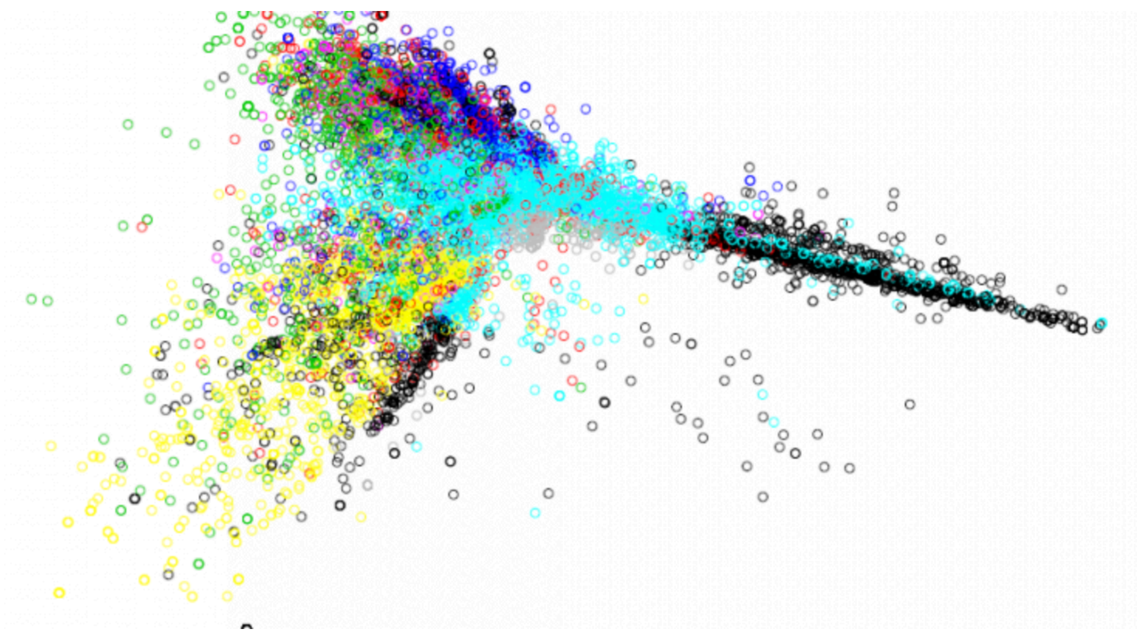
Graph Coloring

- **Input**: Graph $G = (V, E)$ and integer k



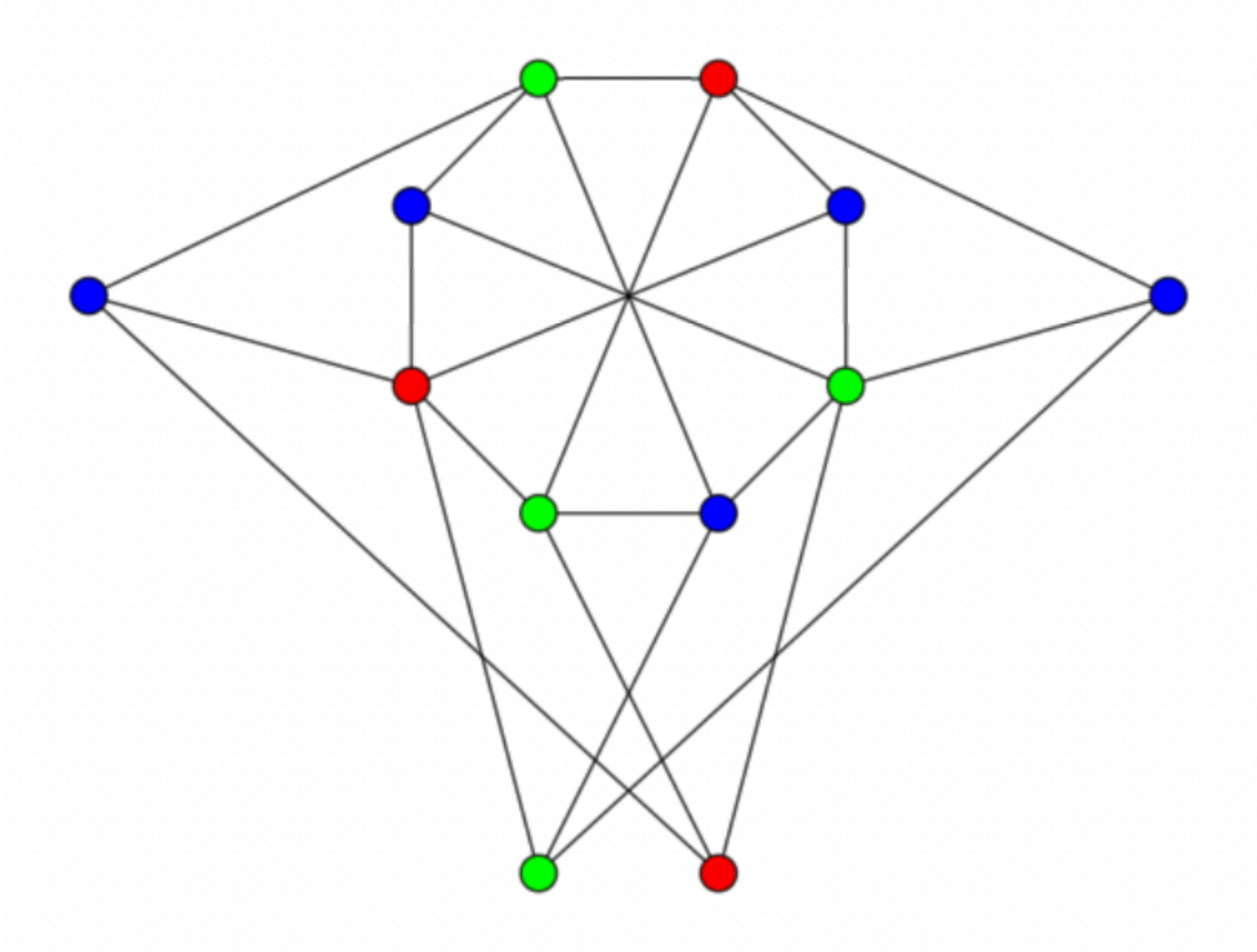
k-means Clustering

- **Input**: Points $P \subset \mathbb{R}^d$ and integer k



Graph Coloring

- **Input**: Graph $G = (V, E)$ and integer k
- **Output**: $V := V_1 \dot\cup \cdots \dot\cup V_k$, such that V_i is an independent set

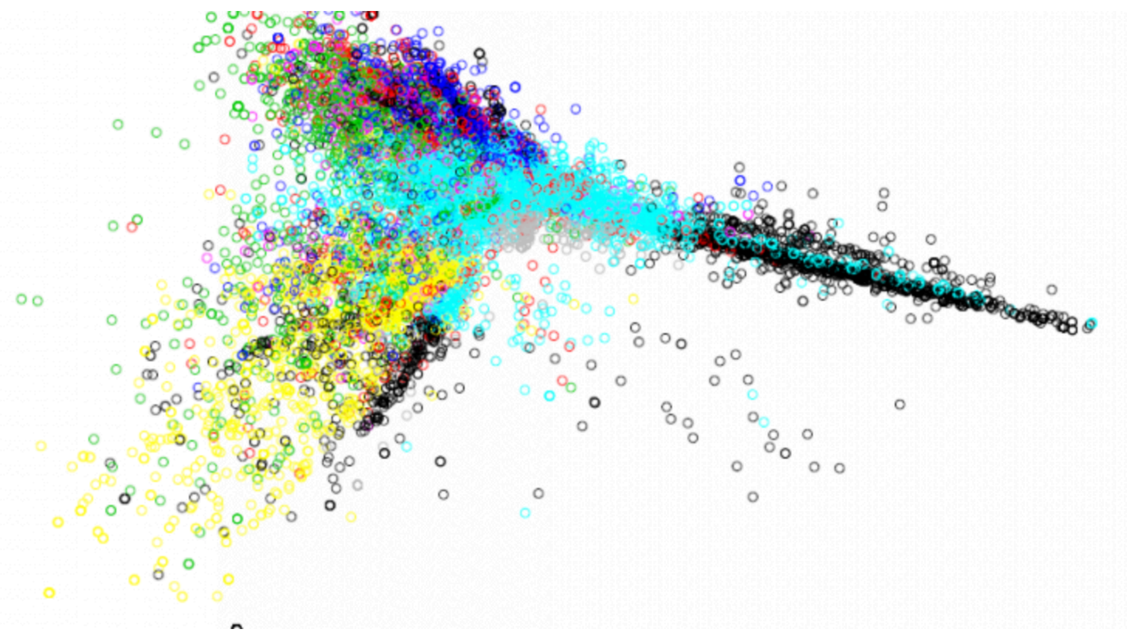


k-means Clustering

- **Input**: Points $P \subset \mathbb{R}^d$ and integer k
- **Output**: $P := P_1 \dot\cup \cdots \dot\cup P_k$, such that

$$\sum_{i \in [k]} \sum_{p \in P_i} \|p - c_i\|_2^2 \text{ is minimized,}$$

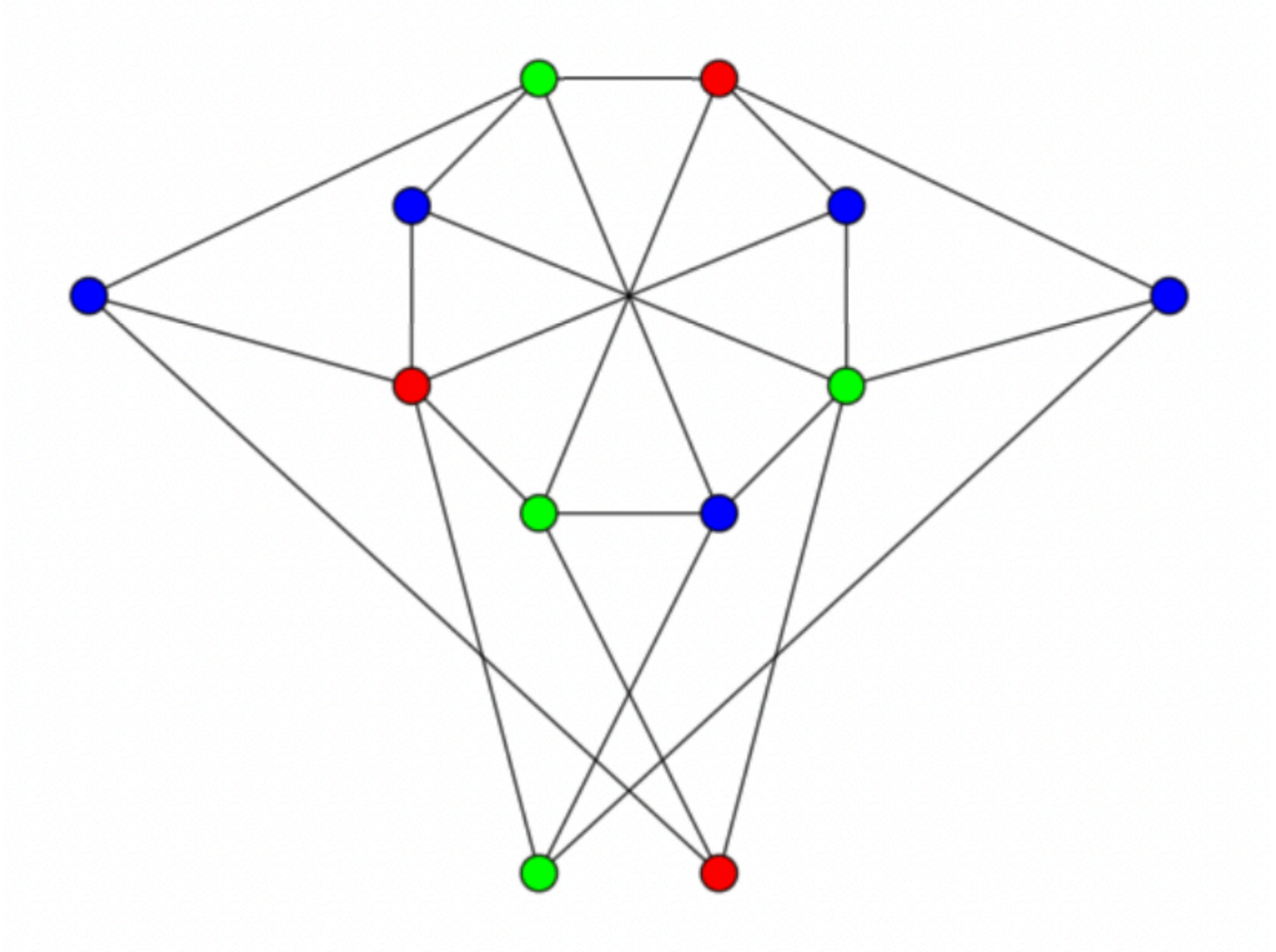
$$\text{where } c_i = \sum_{p \in P_i} \frac{p}{|P_i|}$$



Graph Coloring

- **Input**: Graph $G = (V, E)$ and integer k
- **Output**: $V := V_1 \dot{\cup} \cdots \dot{\cup} V_k$, such that

  V_i is an independent set
- Classic NP-hard problem



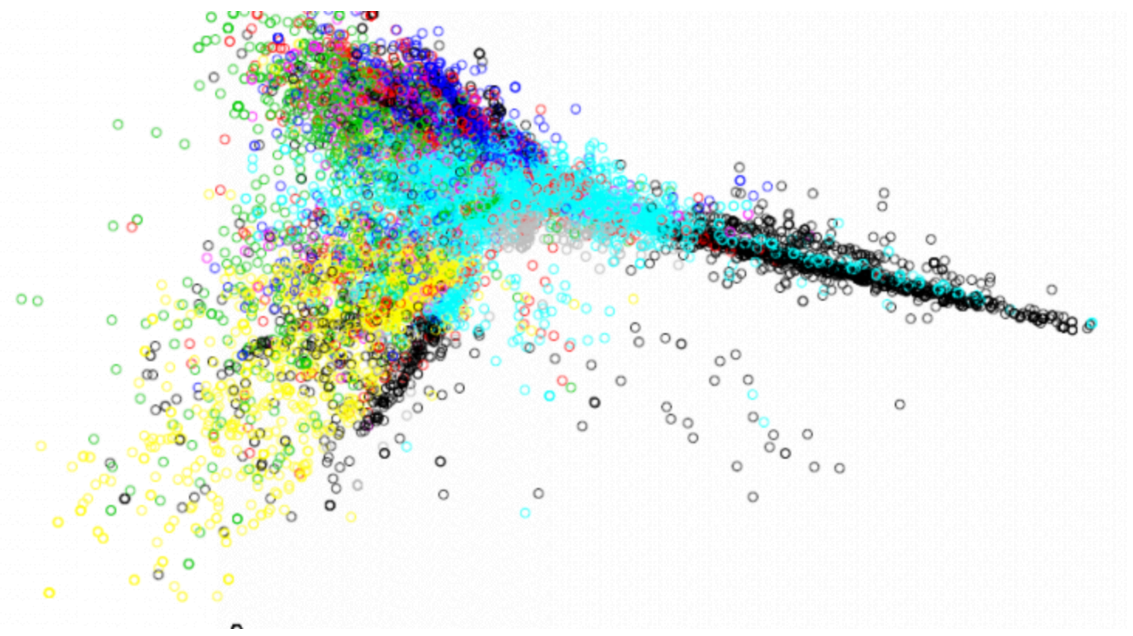
k-means Clustering

- **Input**: Points $P \subset \mathbb{R}^d$ and integer k
- **Output**: $P := P_1 \dot{\cup} \cdots \dot{\cup} P_k$, such that

$$\sum_{i \in [k]} \sum_{p \in P_i} \|p - c_i\|_2^2 \text{ is minimized,}$$

$$\text{where } c_i = \sum_{p \in P_i} \frac{p}{|P_i|}$$

- Important NP-hard problem



k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

k-means Clustering

- Input: Points $P \subset \mathbb{R}^d$

k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

k-means Clustering

- Input: Points $P \subset \mathbb{R}^d$

Orient E arbitrarily

$$v \in V \longrightarrow p_v \in \mathbb{R}^{|E|}$$

$$p_v(e) = \begin{cases} +1 & \text{if } e \text{ is outgoing edge of } v \\ -1 & \text{if } e \text{ is incoming edge of } v \\ 0 & \text{otherwise} \end{cases}$$

k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

k-means Clustering

- Input: Points $P \subset \mathbb{R}^d$

$\{u, v\} \in E \implies$

$\|p_v - p_u\|_2^2 = 4 + 2(d-1) = 2d + 2$

Orient E arbitrarily

$$v \in V \longrightarrow p_v \in \mathbb{R}^{|E|}$$

$$p_v(e) = \begin{cases} +1 & \text{if } e \text{ is outgoing edge of } v \\ -1 & \text{if } e \text{ is incoming edge of } v \\ 0 & \text{otherwise} \end{cases}$$

k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

$\{u, v\} \in E \implies$

$\|p_v - p_u\|_2^2 = 4 + 2(d-1) = 2d + 2$

k-means Clustering

- Input: Points $P \subset \mathbb{R}^d$

$\{u, v\} \notin E \implies$

$\|p_v - p_u\|_2^2 = 2d$

Orient E arbitrarily

$$v \in V \longrightarrow p_v \in \mathbb{R}^{|E|}$$

$$p_v(e) = \begin{cases} +1 & \text{if } e \text{ is outgoing edge of } v \\ -1 & \text{if } e \text{ is incoming edge of } v \\ 0 & \text{otherwise} \end{cases}$$

k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

k-means Clustering

- Input: Points $P \subset \mathbb{R}^{|E|}$

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u, v\} \notin E \\ 2d + 2 & \text{if } \{u, v\} \in E \end{cases}$$

k-coloring to k-means

A simple reduction

Graph k-Coloring

- Input: Graph $G = (V, E)$, G is d-regular

k-means Clustering

- Input: Points $P \subset \mathbb{R}^{|E|}$

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u, v\} \notin E \\ 2d + 2 & \text{if } \{u, v\} \in E \end{cases}$$

$\sum_{i \in [k]} \sum_{p \in P_i} \|p - c_i\|_2^2$ is minimized,

where $c_i = \sum_{p \in P_i} \frac{p}{|P_i|}$

$\Longleftrightarrow$

$\sum_{i \in [k]} \frac{1}{2|P_i|} \sum_{p,q \in P_i} \|p - q\|_2^2$

is minimized,

k-coloring to k-means

A simple reduction

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u,v\} \notin E \\ 2d+2 & \text{if } \{u,v\} \in E \end{cases}$$

Completeness

Soundness

k-coloring to k-means

A simple reduction

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u,v\} \notin E \\ 2d+2 & \text{if } \{u,v\} \in E \end{cases}$$

Completeness

- $V := V_1 \dot\cup \cdots \dot\cup V_k$, such that
  V_i is an independent set

- $\sum_{i\in[k]} \frac{1}{2|P_i|} \sum_{p,q\in P_i} \|p-q\|_2^2 = \sum_{i\in[k]} \frac{1}{2|P_i|} \sum_{p,q\in P_i} 2d$

  $= \sum_{i\in[k]} d \cdot (|P_i| - 1) = d \cdot (|V| - k)$

Soundness

k-coloring to k-means

A simple reduction

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u,v\} \notin E \\ 2d+2 & \text{if } \{u,v\} \in E \end{cases}$$

Completeness

- $V := V_1 \dot\cup \cdots \dot\cup V_{k'}$ such that
  V_i is an independent set
- $$\sum_{i\in[k]} \frac{1}{2|P_i|} \sum_{p,q\in P_i} \|p-q\|_2^2 = \sum_{i\in[k]} \frac{1}{2|P_i|} \sum_{p,q\in P_i} 2d$$
  $$= \sum_{i\in[k]} d\cdot(|P_i|-1) = d\cdot(|V|-k)$$

Soundness

- $P := P_1 \dot\cup \cdots \dot\cup P_k$ is some clustering
- $$\forall i \in [k], \sum_{p,q\in P_i} \|p-q\|_2^2 \geq 2d\cdot|P_i|\cdot(|P_i|-1)$$
- $$\exists i \in [k], \sum_{p,q\in P_i} \|p-q\|_2^2 > 2d\cdot|P_i|\cdot(|P_i|-1)$$

k-coloring to k-means

A simple reduction

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u, v\} \notin E \\ 2d + 2 & \in E \end{cases}$$

3-coloring is NP-hard

$\Longrightarrow$

3-means is NP-hard

Completenes

- $V := V_1 \dot\cup \cdots \dot\cup V_k$, such that
  V_i is an independent set
- $\displaystyle\sum_{i \in [k]} \frac{1}{2|P_i|} \sum_{p,q \in P_i} \|p - q\|_2^2 = \sum_{i \in [k]} \frac{1}{2|P_i|} \sum_{p,q \in P_i} 2d$
  $\displaystyle= \sum_{i \in [k]} d \cdot (|P_i| - 1) = d \cdot (|V| - k)$

Soundness

- $\dot\cup \cdots \dot\cup P_k$ is some clustering
- $\displaystyle\forall i \in [k], \sum_{p,q \in P_i} \|p - q\|_2^2 \geq 2d \cdot |P_i| \cdot (|P_i| - 1)$
- $\displaystyle\exists i \in [k], \sum_{p,q \in P_i} \|p - q\|_2^2 > 2d \cdot |P_i| \cdot (|P_i| - 1)$

What about 1-means?

What about 2-means?

- 2-coloring reduces to 2-means
- 2-coloring is in P so 2-means is in P

What about 2-means?

- 2-coloring reduces to 2-means
- 2-coloring is in P so 2-means is in P

What about 2-means?

- 2-means is NP-hard [Dasgupta-Freund’09]

What about 2-means?

- 2-means is NP-hard [Dasgupta-Freund'09]

- **Proof Strategy**:

  (i) Structured 3-NAE-SAT is NP-hard

  (ii) NAE-SAT is reduced to distance matrix of points of 2-means instance

  (iii) Distance matrix can be realized in Euclidean space (PSD check)

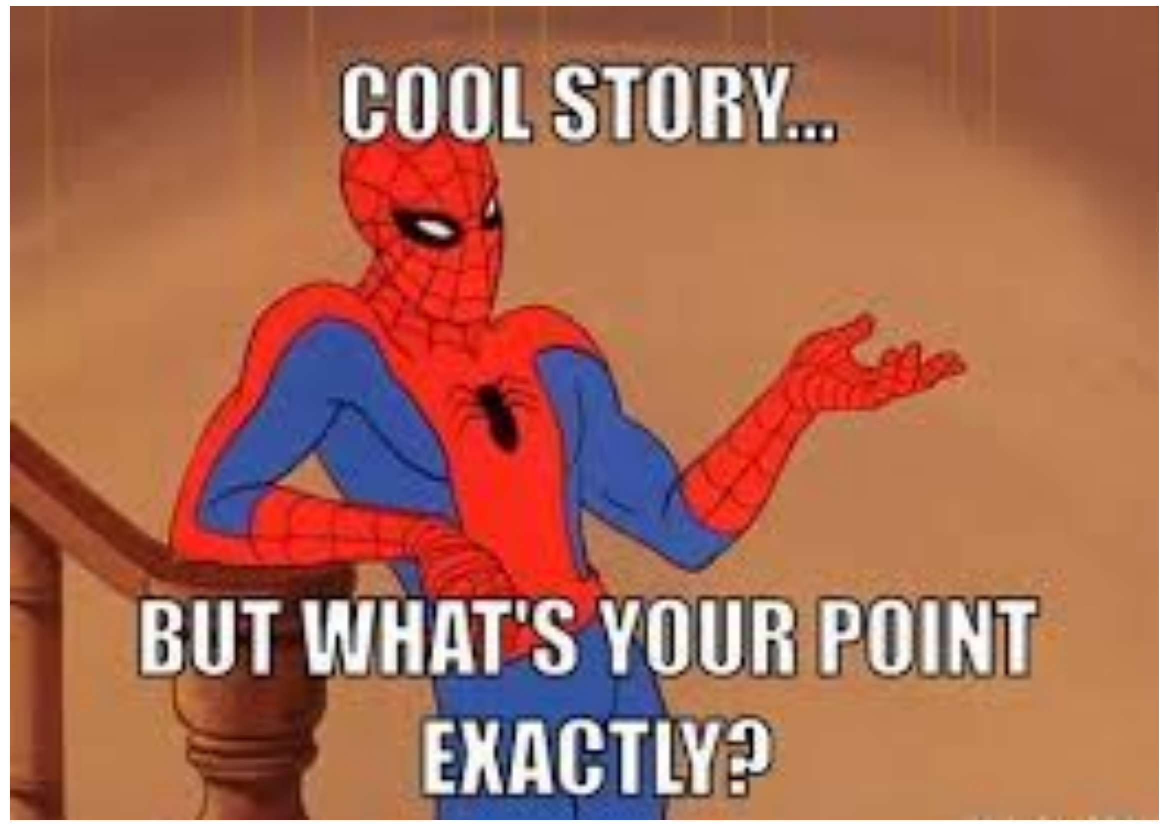
What about 2-means?

- 2-means is NP-hard [Dasgupta-Freund'09]
- **Proof Strategy**:

  (i) Structured 3-NAE-SAT is NP-hard

  (ii) NAE-SAT is reduced to distance matrix of points of 2-means instance

  (iii) Distance matrix can be realized in Euclidean space (PSD check)



Almost 2-coloring to 2-means

New Reduction

Almost 2-coloring to 2-means

New Reduction

Balanced Max-Cut

- Input: Graph $G = (V, E)$, G is d-regular
- Output: $V := V_1 \dot\cup V_2$, to minimize:

$$|E(V_1, V_2)| \cdot \left(1 + \frac{||V_1| - |V_2||}{|V|}\right)$$

2-means Clustering

- Input: Points $P \subset \mathbb{R}^d$
- Output: $P := P_1 \dot\cup P_2$, to minimize:

$$\frac{\sum_{p,q \in P_1} \|p - q\|_2^2}{2|P_1|} + \frac{\sum_{p,q \in P_2} \|p - q\|_2^2}{2|P_2|}$$

Almost 2-coloring to 2-means

New Reduction

Balanced Max-Cut

- Input: Graph $G = (V, E)$, G is d-regular
- Output: $V := V_1 \dot\cup V_2$, to minimize:

$$|E(V_1, V_2)| \cdot \left(1 + \frac{||V_1| - |V_2||}{|V|}\right)$$

- We give NP-hardness reduction from Linear 4-regular 3-NAE-SAT

2-means Clustering

- Input: Points $P \subset \mathbb{R}^d$
- Output: $P := P_1 \dot\cup P_2$, to minimize:

$$\frac{\sum_{p,q \in P_1} \|p - q\|_2^2}{2|P_1|} + \frac{\sum_{p,q \in P_2} \|p - q\|_2^2}{2|P_2|}$$

Almost 2-coloring to 2-means

New Reduction

Balanced Max-Cut

- Input: Graph $G = (V, E)$, G is d-regular
- Output: $V := V_1 \dot\cup V_2$, to minimize:

$$|E(V_1, V_2)| \cdot \left(1 + \frac{||V_1| - |V_2||}{|V|}\right)$$

- We give NP-hardness reduction from Linear 4-regular 3-NAE-SAT

Orient E arbitrarily

$$v \in V \longrightarrow p_v \in \mathbb{R}^{|E|}$$

$$p_v(e) = \begin{cases} +1 & \text{if } e \text{ is outgoing edge of } v \\ -1 & \text{if } e \text{ is incoming edge of } v \\ 0 & \text{otherwise} \end{cases}$$

$$\|p_u - p_v\|_2^2 = \begin{cases} 2d & \text{if } \{u, v\} \notin E \\ 2d + 2 & \text{if } \{u, v\} \in E \end{cases}$$

Reverse the Reduction

Max-Cut on n vertices
can be solved in
$2^{\omega n/3}$ time

[Williams'05]

Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

2-means on n points can be solved in $2^{\omega n/3}$ time

This paper

Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

→

State-of-the-art

$2^{\omega n/3} \leq 1.73^n$

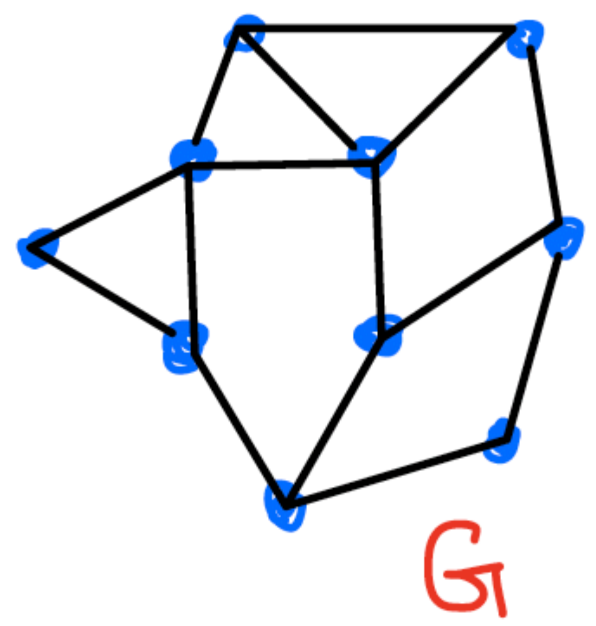
2-means on n points can be solved in $2^{\omega n/3}$ time

This paper

Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

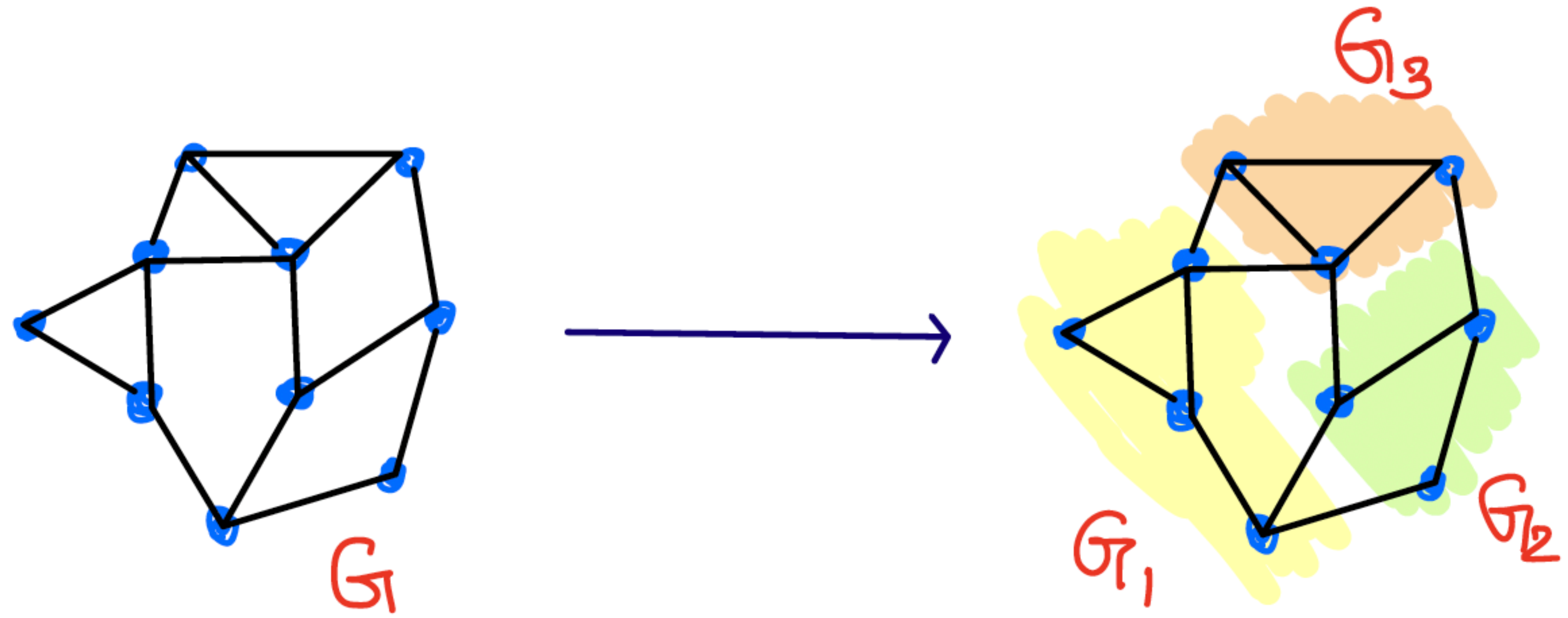
[Williams'05]



Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

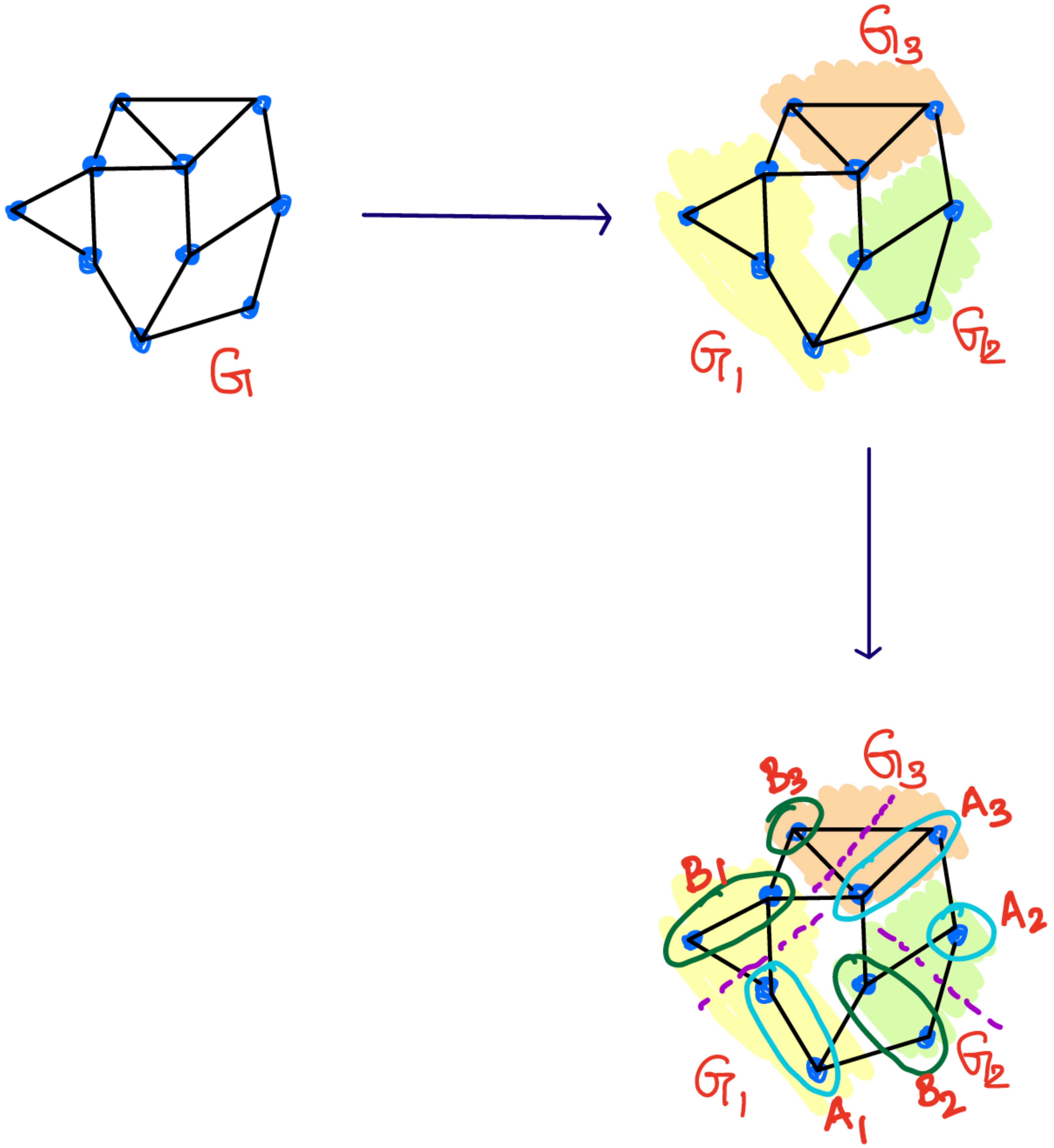


Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

- Enumerate all cuts (A_i, B_i) of G_i

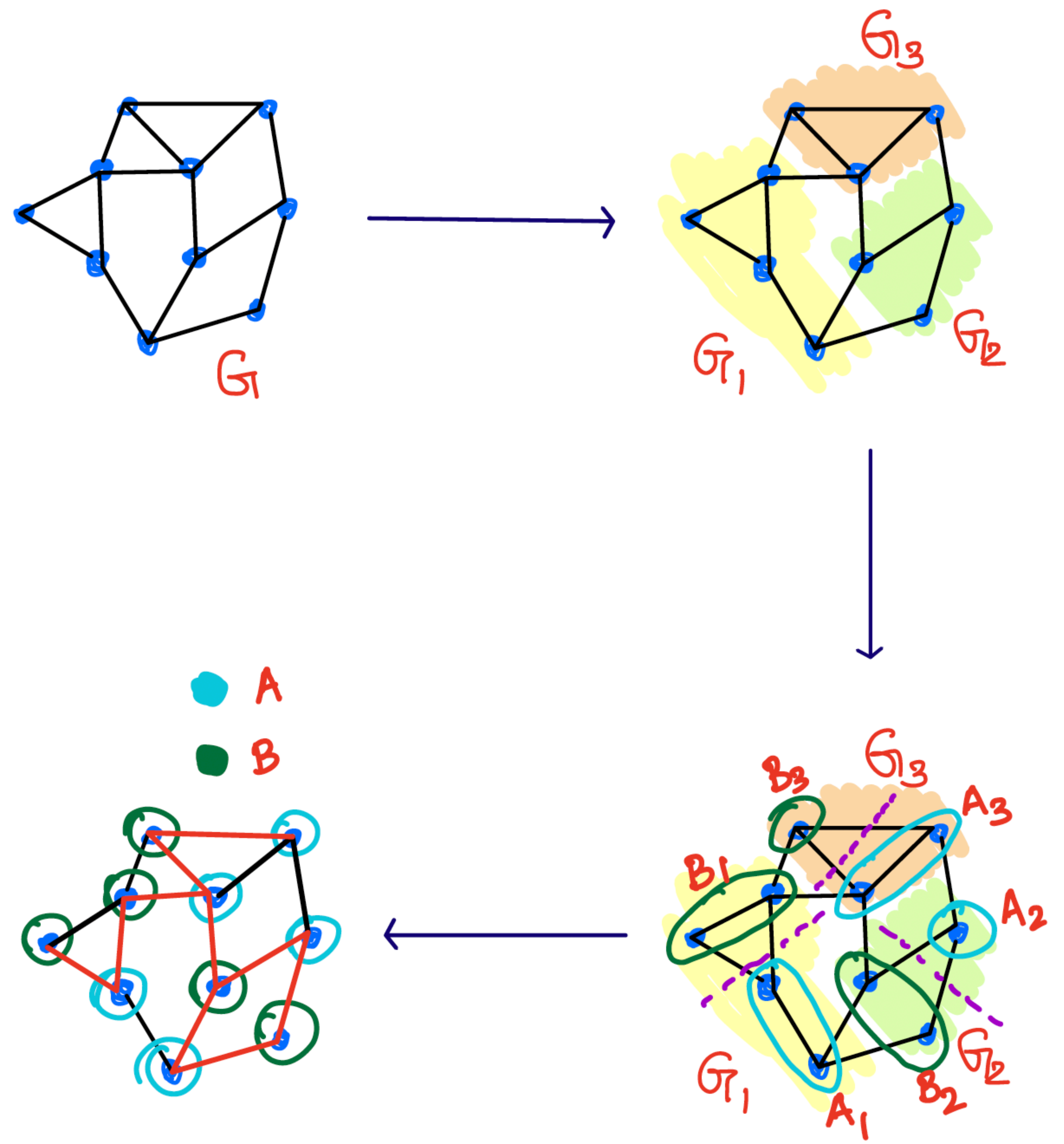


Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

- Enumerate all cuts (A_i, B_i) of G_i
- Construct graph H on $3 \cdot 2^{n/3}$ nodes
- Edge Weight of H is sum of cut edges

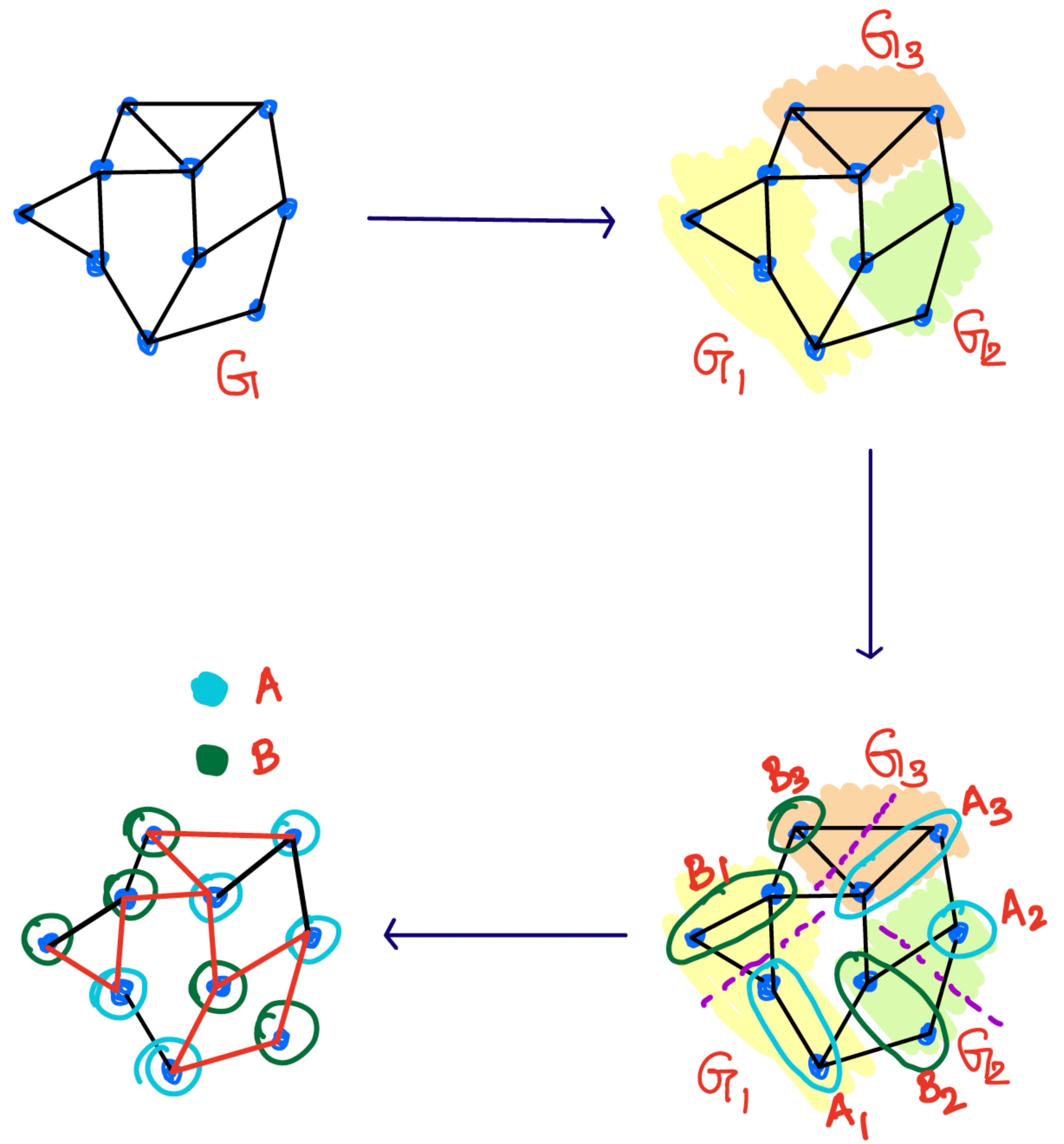


Reverse the Reduction

Max-Cut on n vertices can be solved in $2^{\omega n/3}$ time

[Williams'05]

- Enumerate all cuts (A_i, B_i) of G_i
- Construct graph H on $3 \cdot 2^{n/3}$ nodes
- Edge Weight of H is sum of cut edges
- Solve weighted triangle detection on $3 \cdot 2^{n/3}$ node graph H

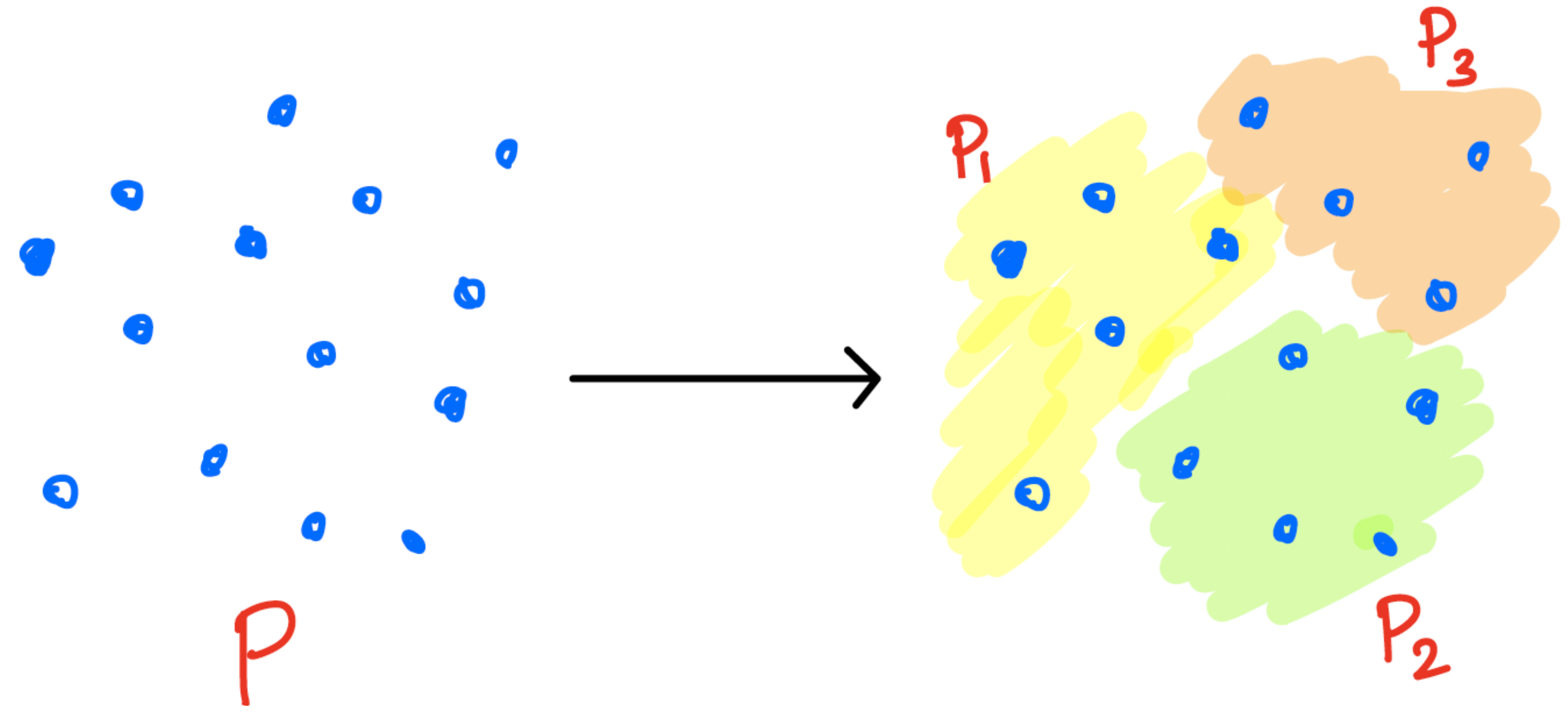


Reverse the Reduction

2-means on n points
can be solved in
$2^{\omega n/3}$ time

Reverse the Reduction

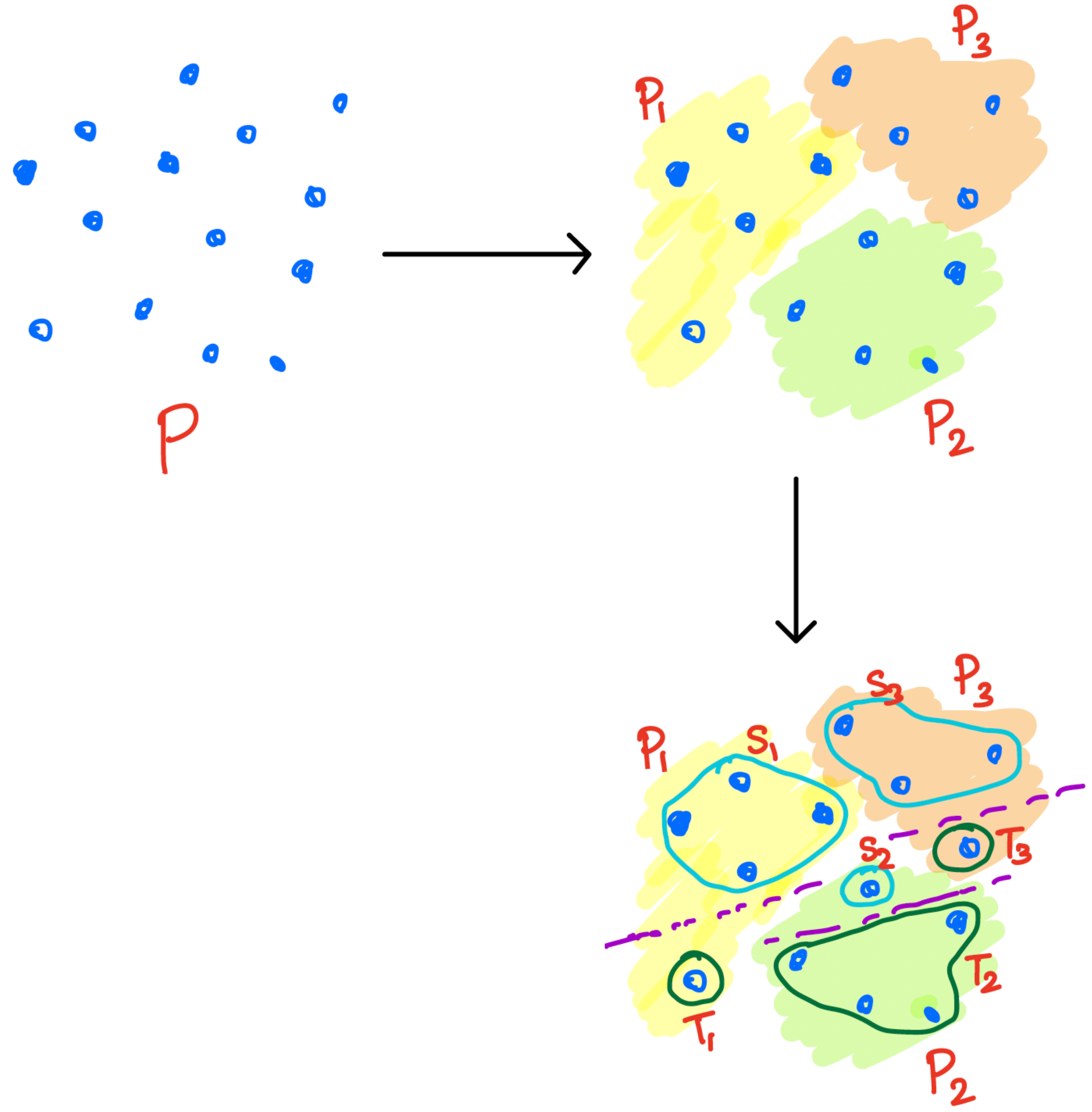
2-means on n points can be solved in $2^{\omega n/3}$ time



Reverse the Reduction

2-means on n points
can be solved in
$2^{\omega n/3}$ time

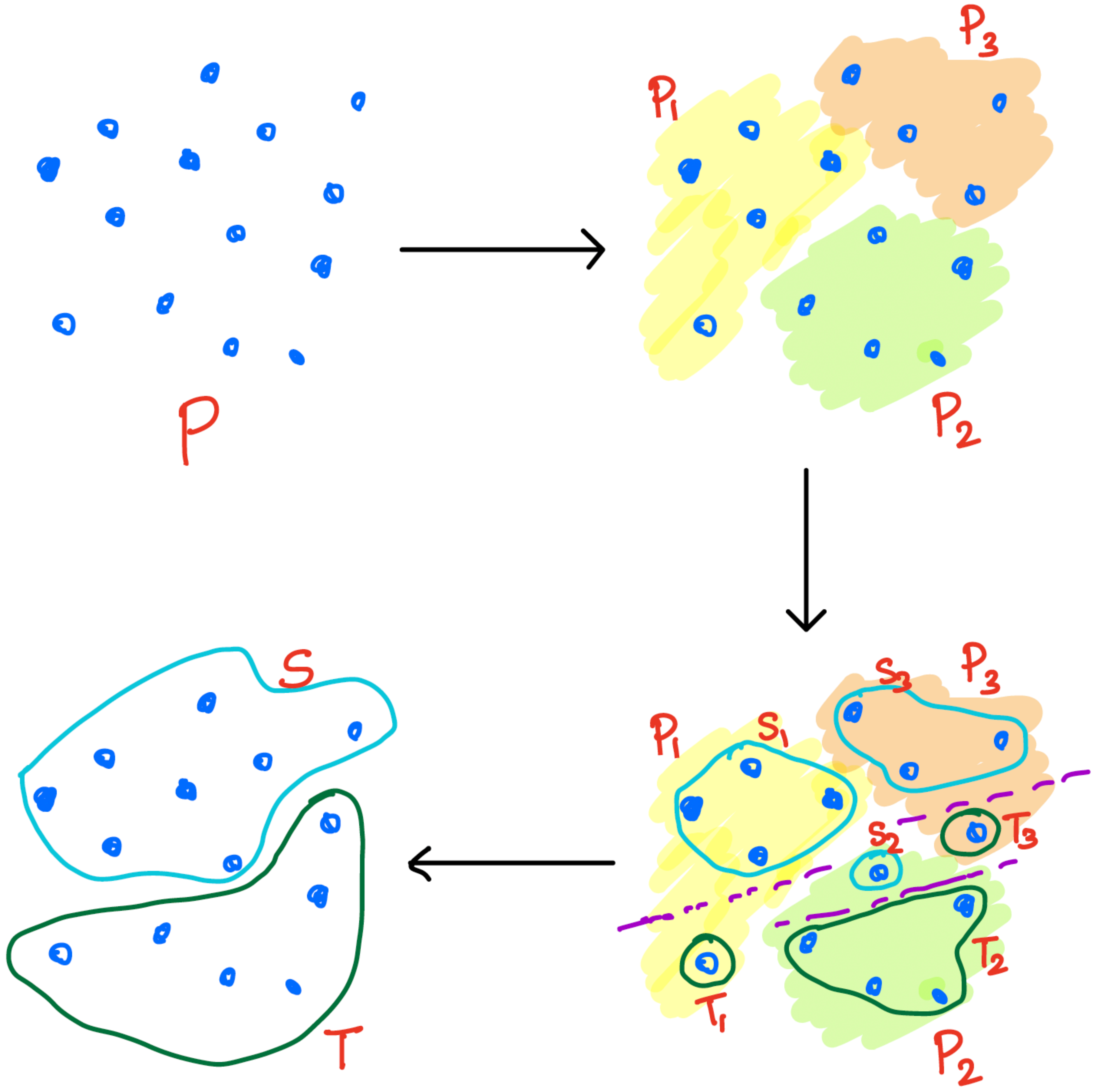
- Enumerate all clustering (S_i, T_i) of P_i



Reverse the Reduction

2-means on n points can be solved in $2^{\omega n/3}$ time

- Enumerate all clustering (S_i, T_i) of P_i
- Construct graph H on $3 \cdot 2^{n/3}$ nodes
- Edge Weight of H is sum of pairwise intracluster distances
- Solve weighted triangle detection on $3 \cdot 2^{n/3}$ node graph H



Open Directions

3-coloring $\Longrightarrow$ **3**-means

Open Directions

3-coloring $\overset{?}{\Longleftarrow\!\!\Longrightarrow}$ **3**-means

Can we use ideas from
3-coloring algorithms
to obtain $(2 - \varepsilon)^n$ time algorithm
for 3-means?

Open Directions

Can we use ideas from
2-means algorithm
to obtain $(2-\varepsilon)^n$ time algorithm
for 2-median or 2-center?

Thank you for engaging!