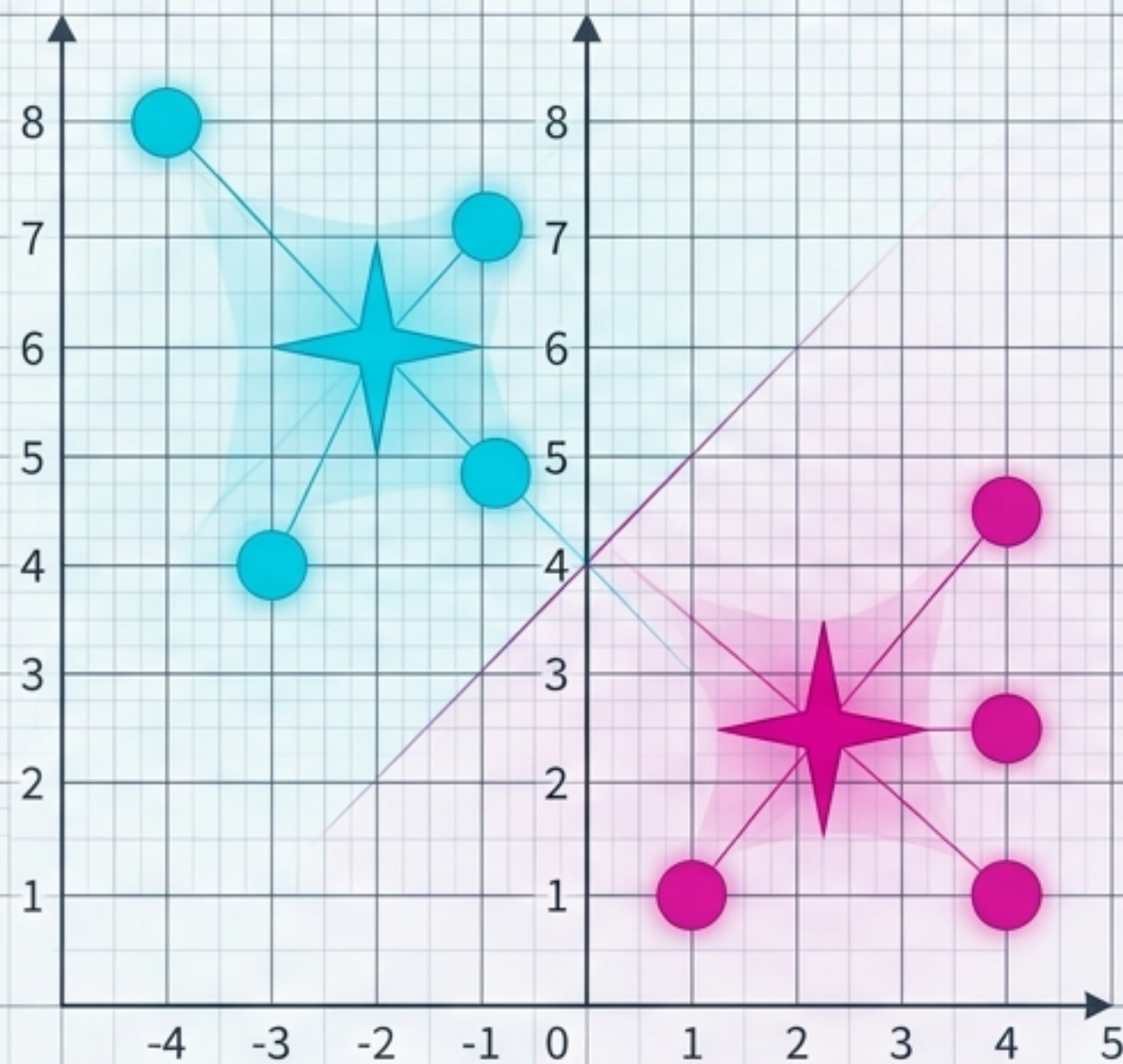


K-means法 完全解剖：7つの点のクラスタリング

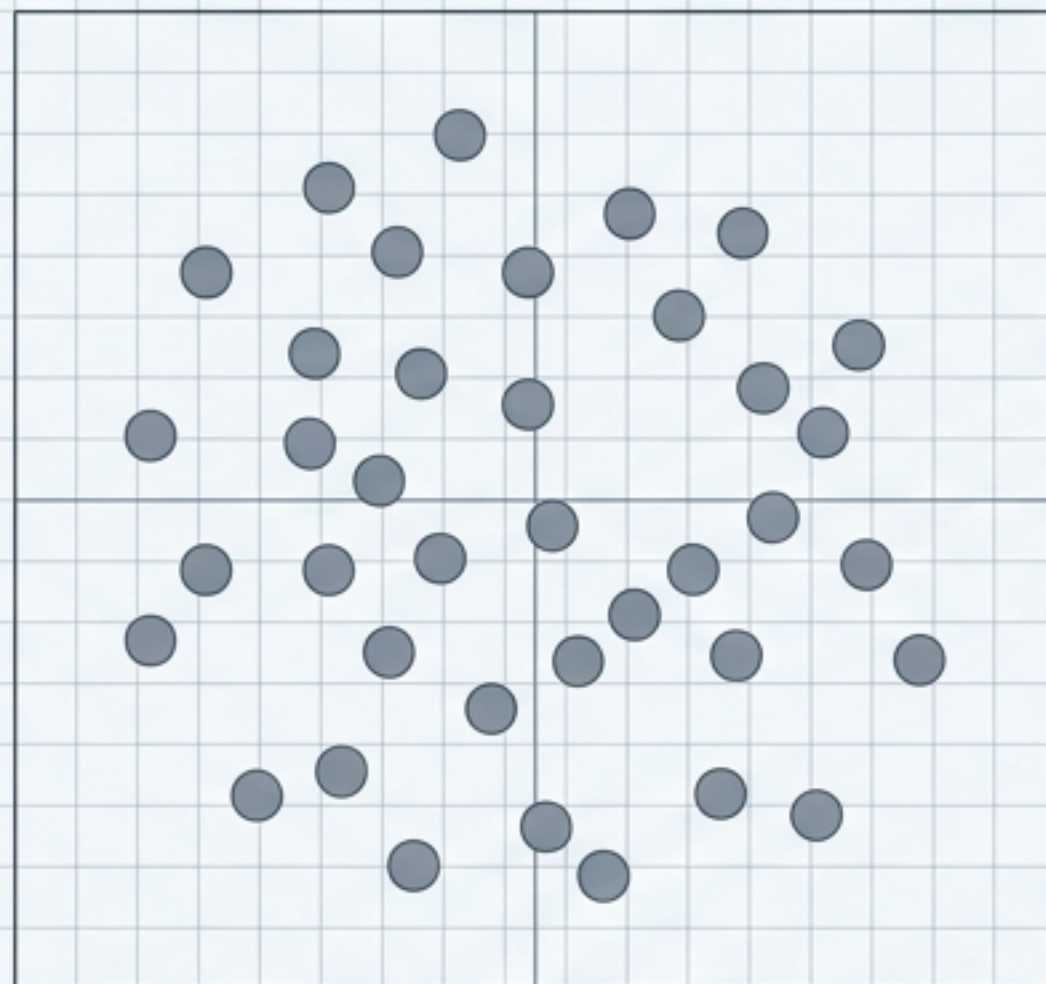
令和3年度 応用情報技術者試験から読み解く、第3ループでの「収束」プロセス



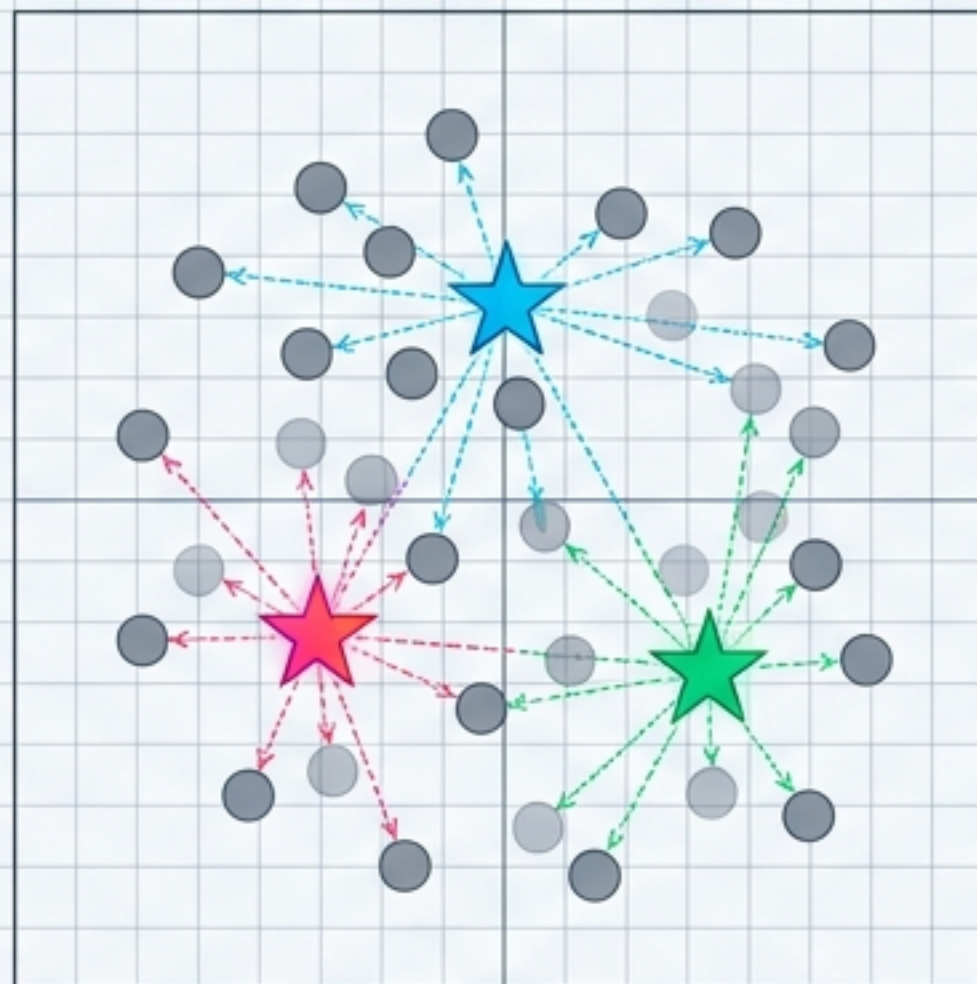
応用情報技術者試験 午後 問3 アルゴリズムの視覚的証明

2021s_K-meansの解説

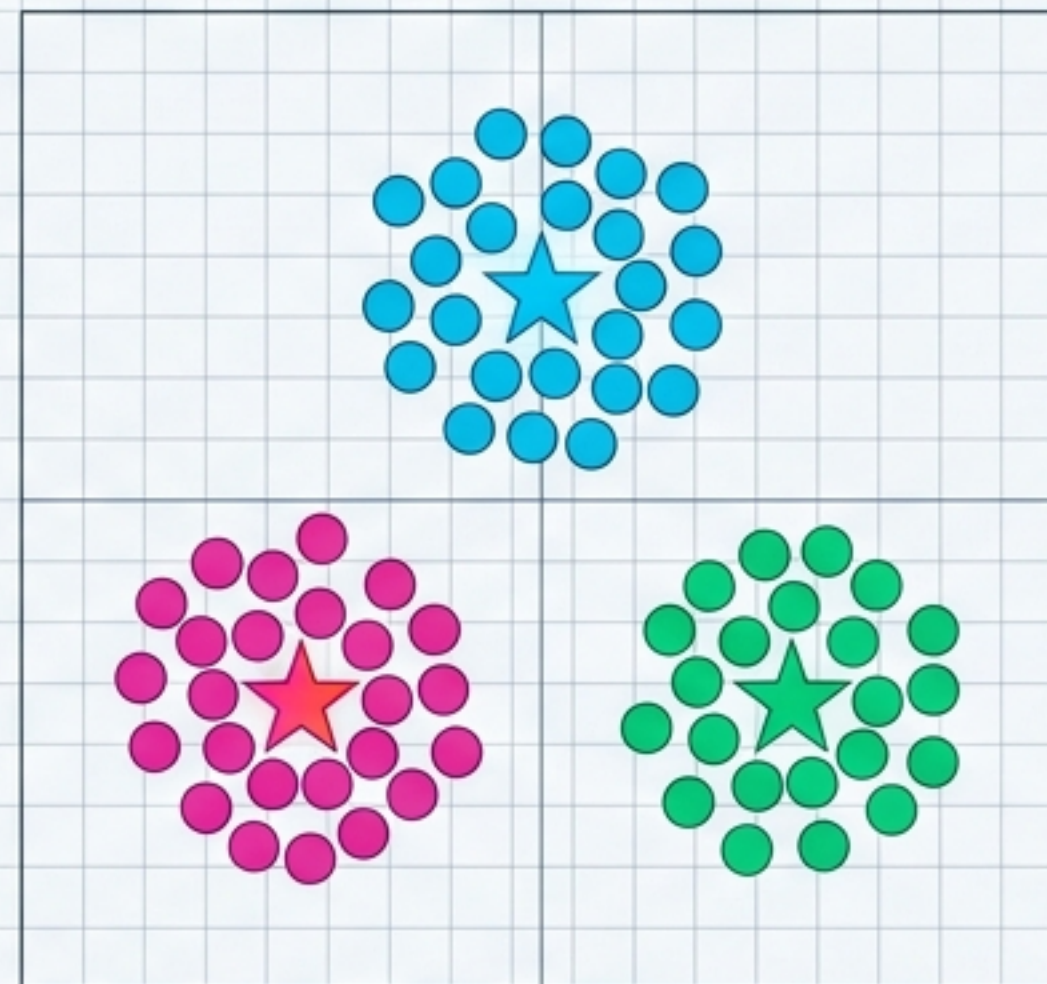
K-means法：データに潜む「グループ」を見つけ出す



Before



Process



After

K-means法の基本概念

目的: 教師なし学習によるデータの自動グループ化 (K個のクラスタに分類)

黄金ルール:

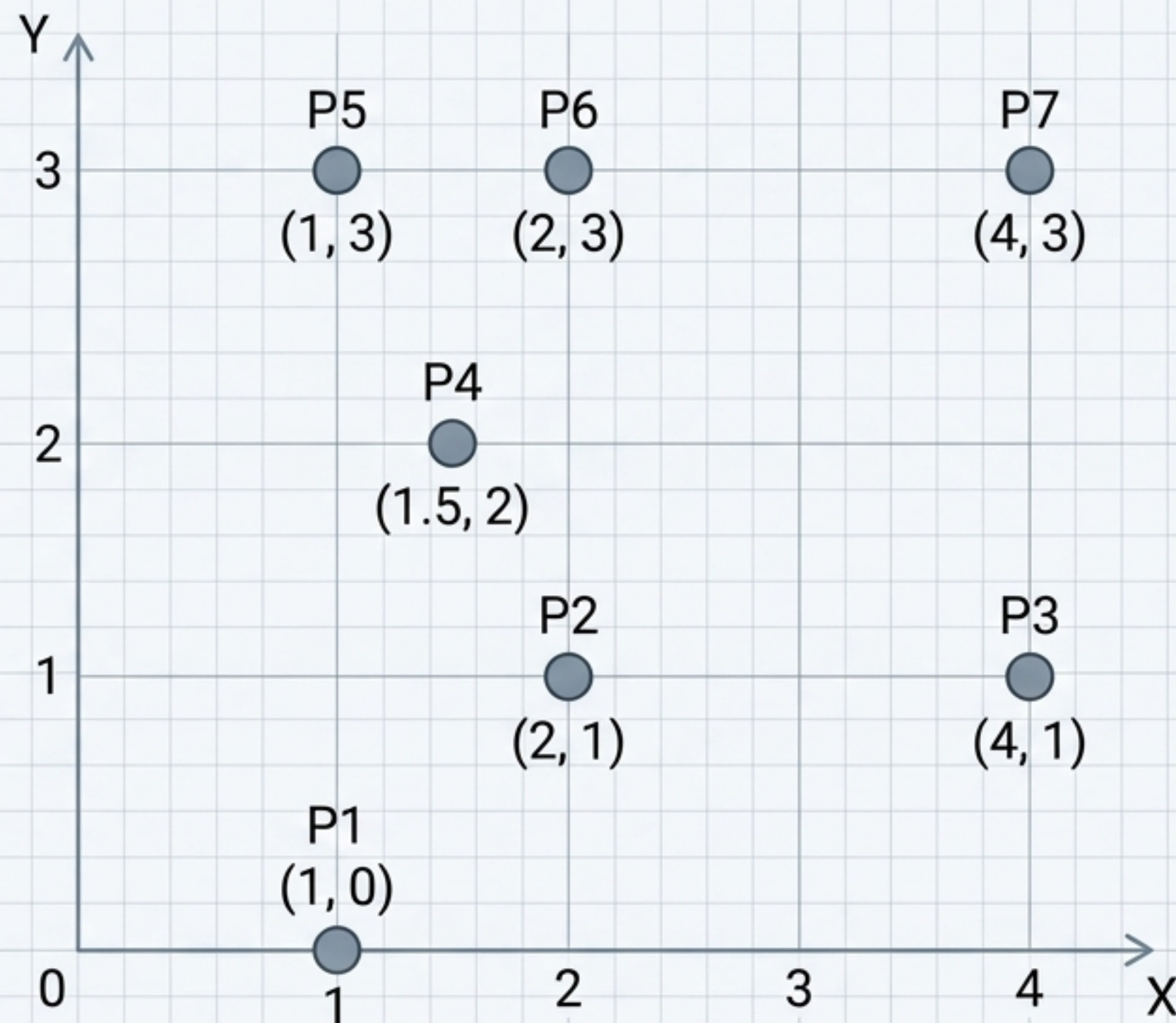
1. 【割当】 各点を最も近い「中心」のグループにする
2. 【更新】 グループの「重心」を新しい中心にする
3. 【収束】 変化がなくなるまで1と2を繰り返す

ミッション：7つの点を2つのクラスタに分類せよ

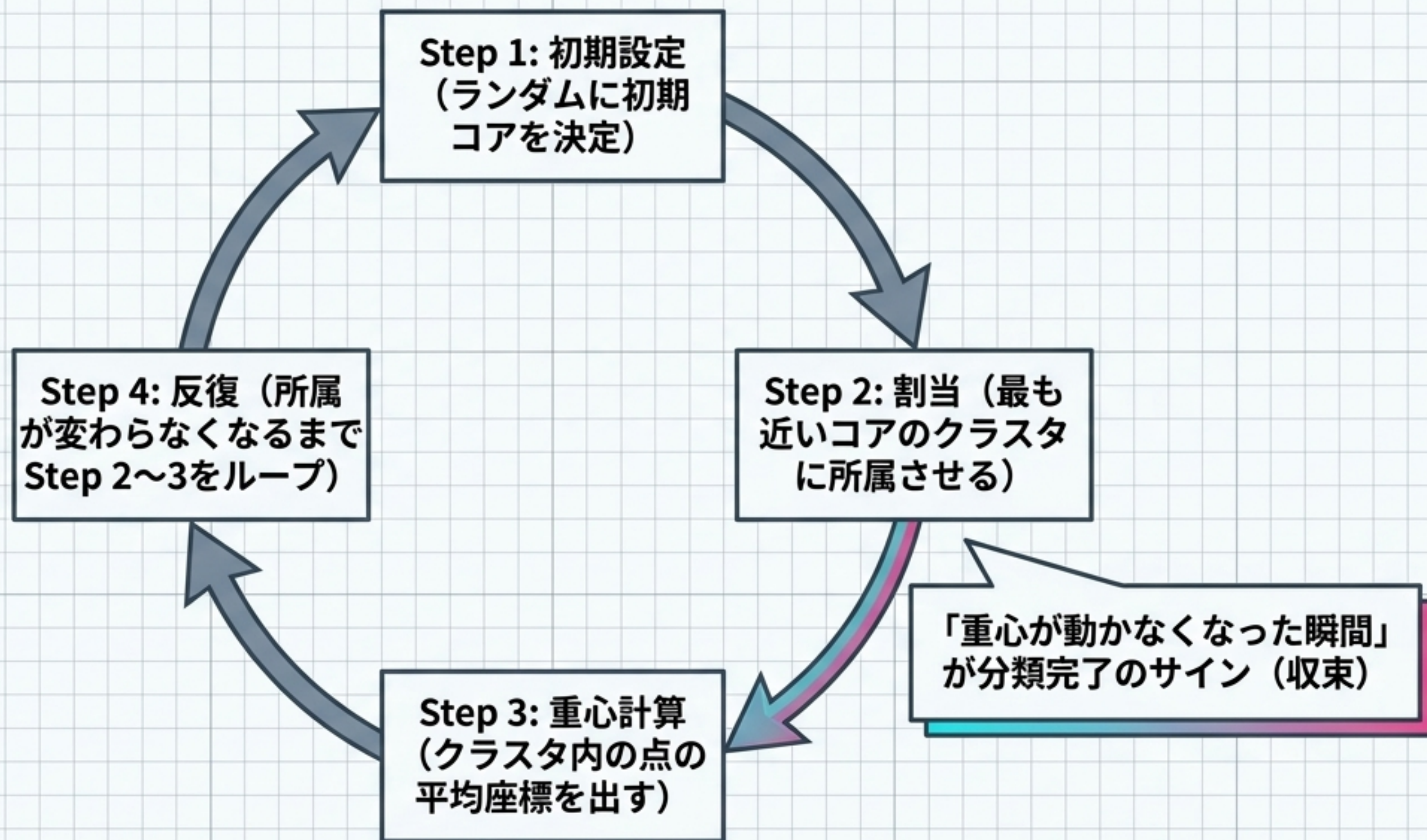
出典: 令和3年度 春期 応用情報技術者試験
午後 問3

設定条件：

- ・ 対象データ：2次元座標上の7点
- ・ 分類数： $K=2$ (2つのグループに分ける)
- ・ 距離の測り方：ユークリッド距離



収束への4つのステップ（ループ構造）



第1ループ：初期コアの選定と最初の分類

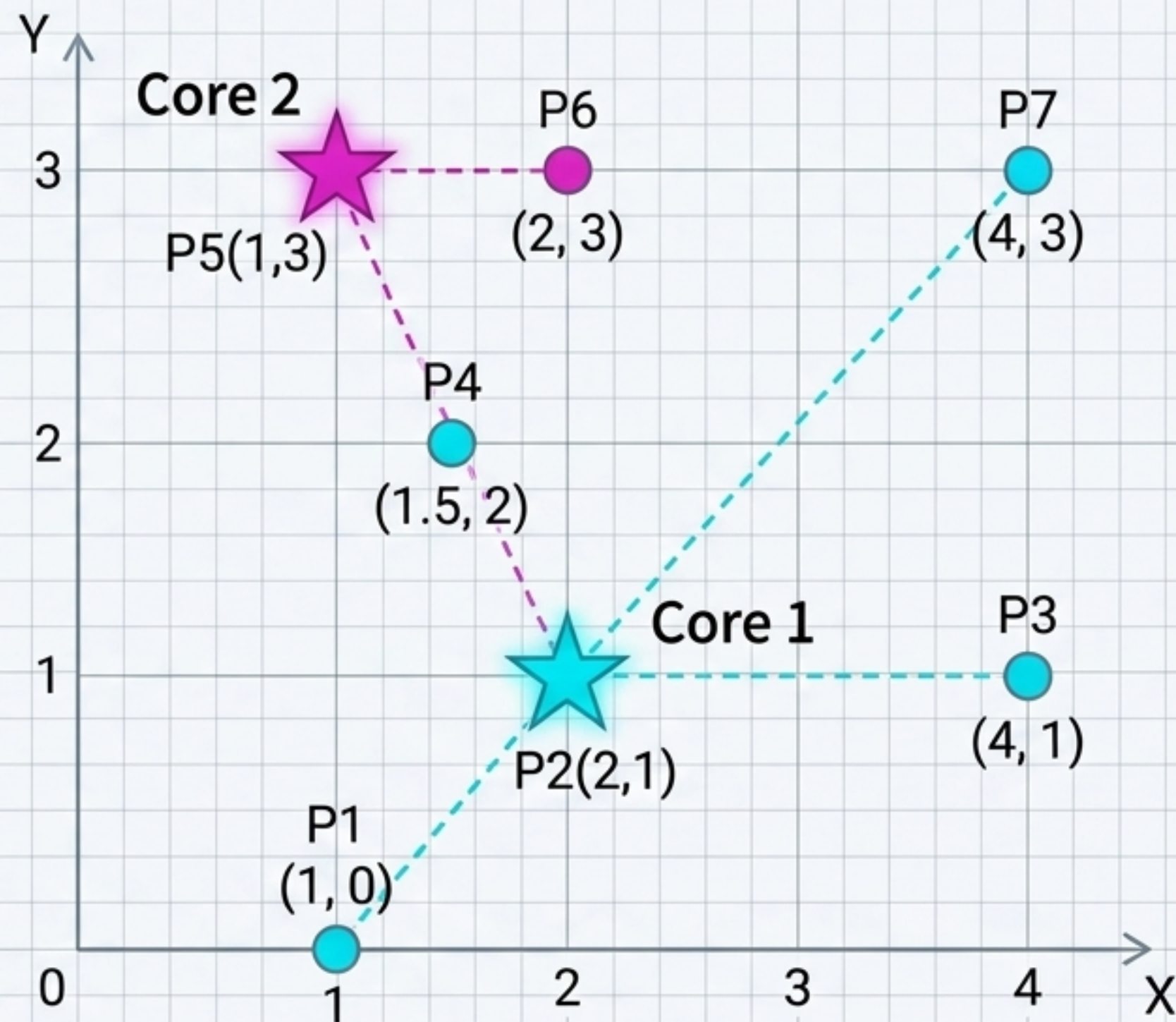
【Step 1 & 2 実行】

初期コア:

- コア1 = P2 (2, 1)
- コア2 = P5 (1, 3)

初期割当結果:

- クラスタ1 (青): P1, P2, P3, P4, P7
- クラスタ2 (赤): P5, P6



第1ループ：集団の「新しい中心」を導き出す

【Step 3 実行】各グループの平均座標を計算

新重心 G1 (クラスタ1):

$$X: (1 + 2 + 4 + 1.5 + 4) \div 5 = 2.5$$

$$Y: (0 + 1 + 1 + 2 + 3) \div 5 = 1.4$$

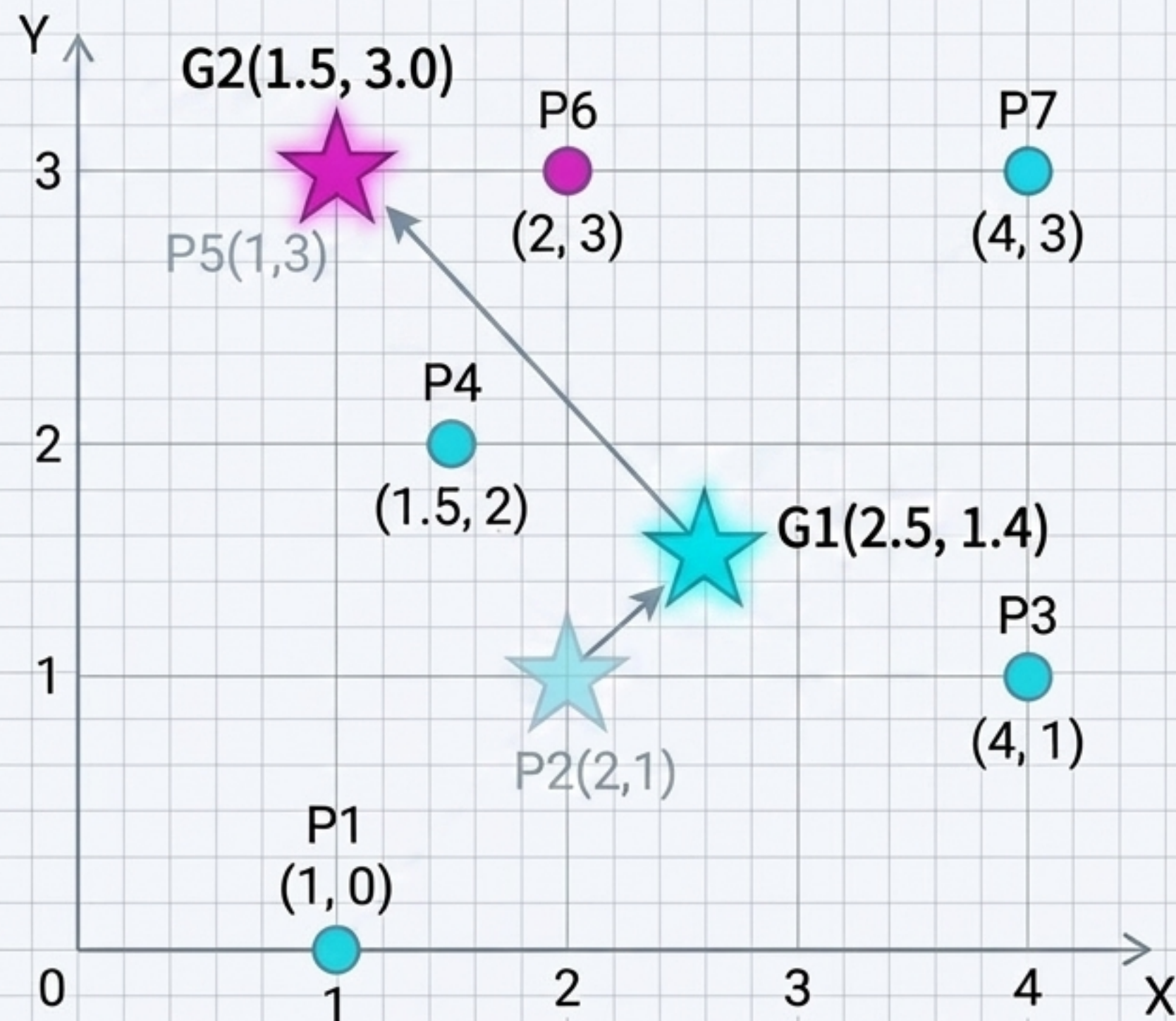
→ 確定: G1 (2.5, 1.4)

新重心 G2 (クラスタ2): ※設問ア

$$X: (1 + 2) \div 2 = 1.5$$

$$Y: (3 + 3) \div 2 = 3.0$$

→ 確定: G2 (1.5, 3.0)



第2ループ：重心の移動がもたらす「所属の変更」

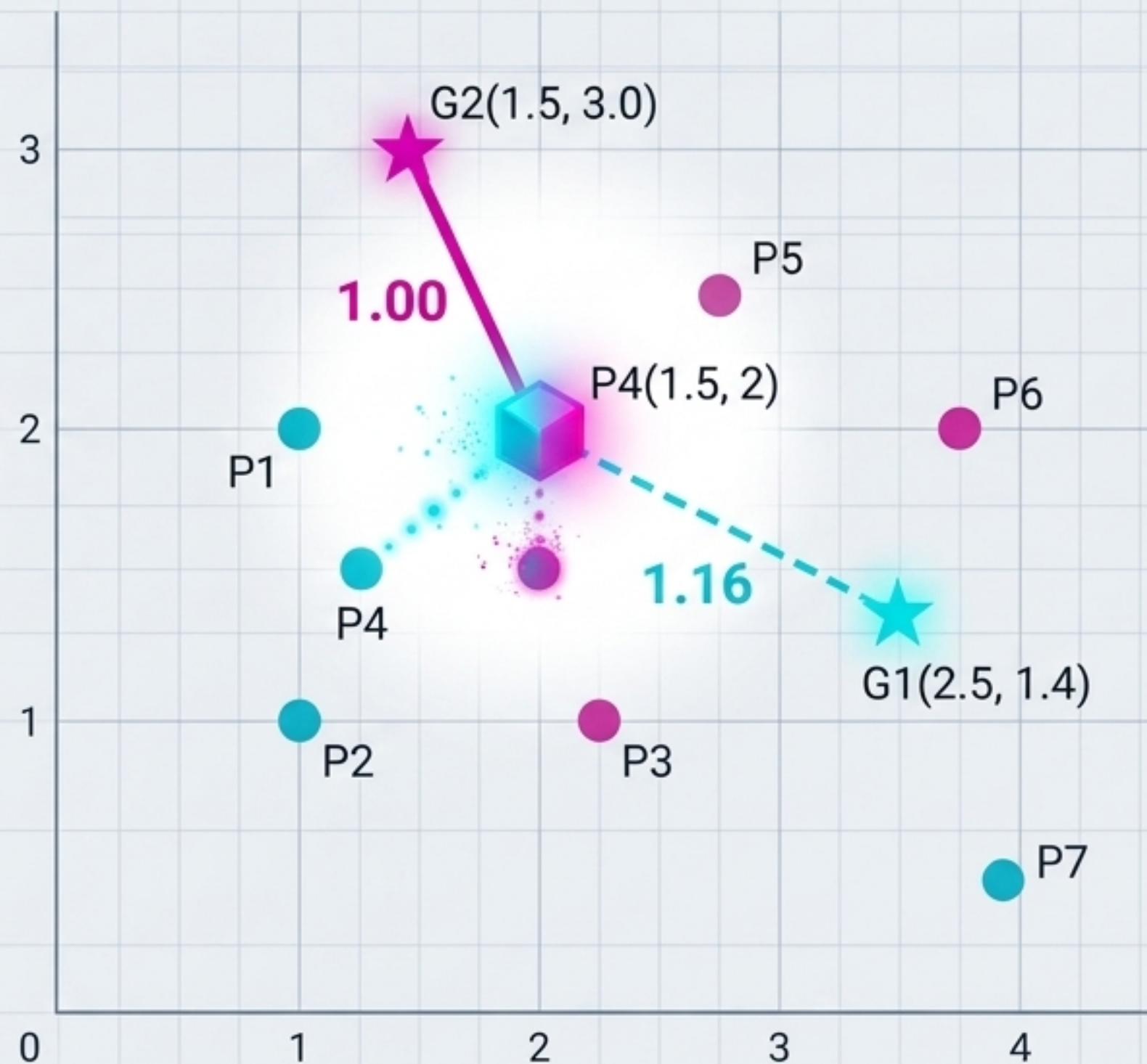
注目の点 P4 (1.5, 2)

重心が動いたことで、各点の距離を再計算します。

旧重心 G1 (2.5, 1.4) までの距離 ≈ 1.16
新重心 G2 (1.5, 3.0) までの距離 = 1.00

結論:

P4 は新しく誕生した G2 の方が
近くなったため、クスタ1(青) から
クスタ2(赤) へ移動！



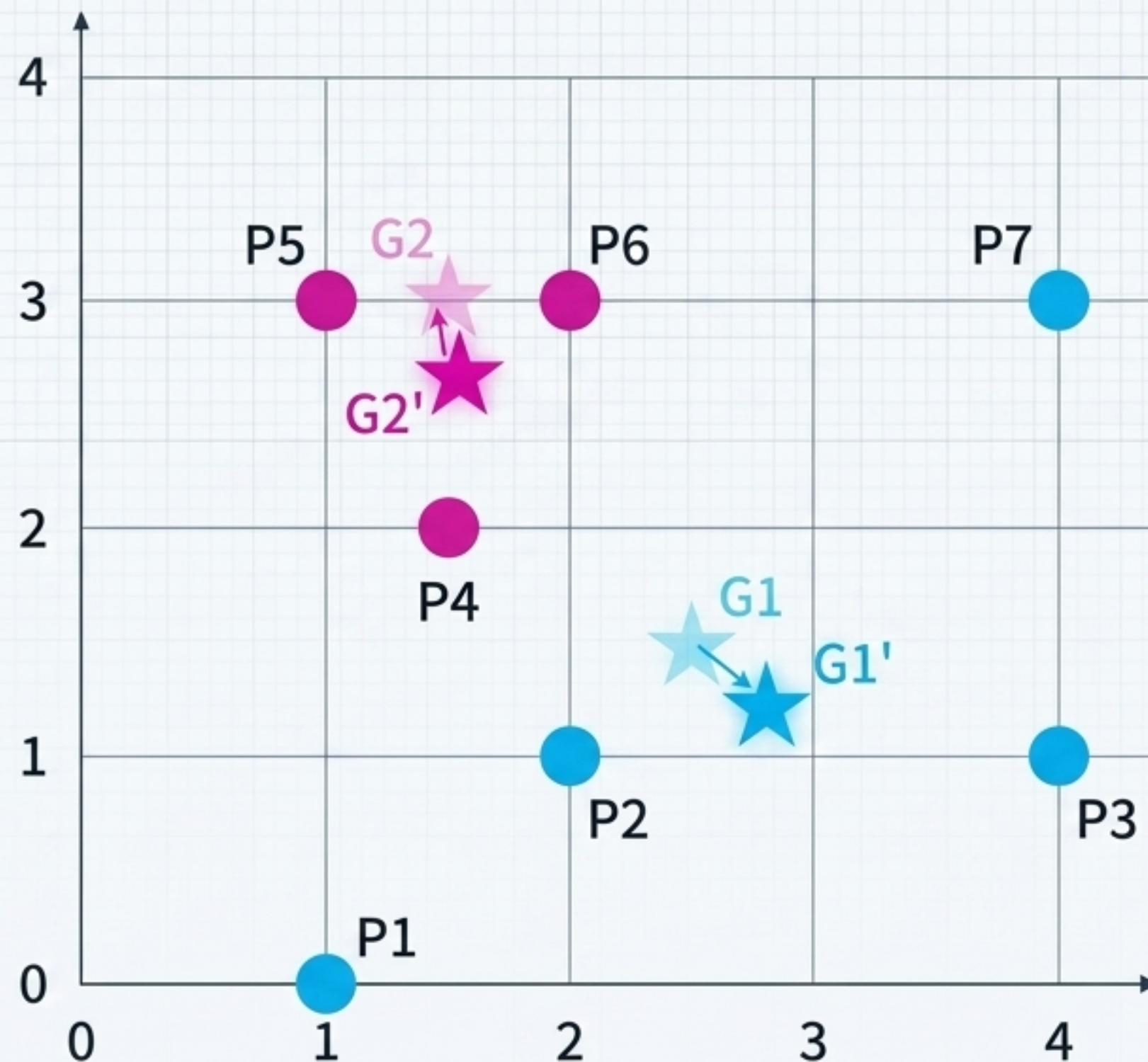
第2ループ：洗練されたクラスタの重心更新

P4 がクラスタ2へ移動したことに伴い、
両グループのメンバー構成が変化しました。

構成員が変わったため、再度重心を計算
し直します。

新クラスタ1: (P1, P2, P3, P7)
➡ 最新重心 $G1'$ (2.75, 1.25)

新クラスタ2: (P4, P5, P6)
➡ 最新重心 $G2'$ (1.5, 2.67)



第3ループ：これ以上動かない「収束」の証明

境界判定による終了確認

最新の重心 $G1'$, $G2'$ に対して、全点の距離を再計算します。

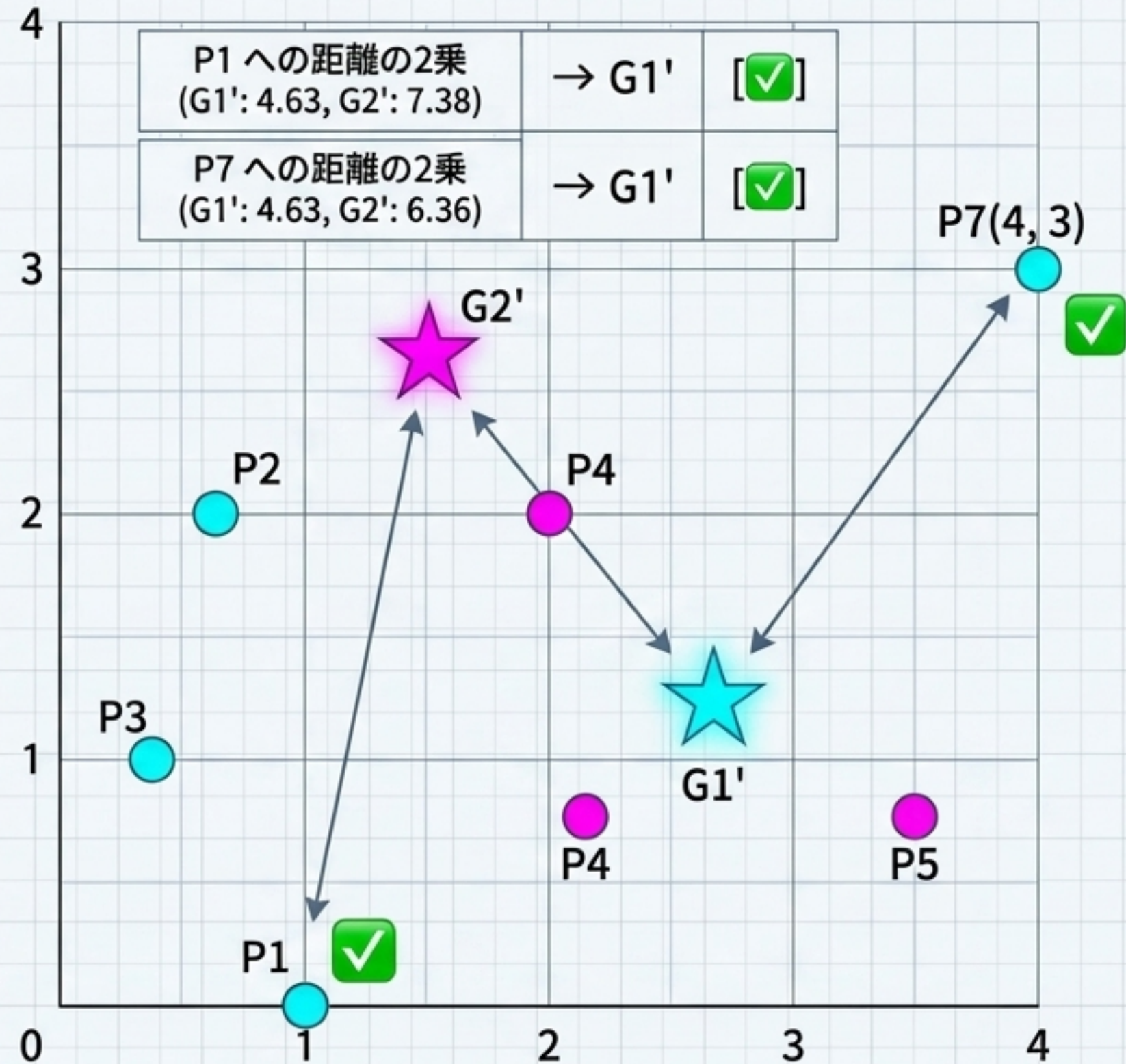
【ボーダーライン $P1(1,0)$ の判定】

- $G1'$ までの距離の2乗 = 4.63 (クラスタ1維持)
- $G2'$ までの距離の2乗 = 7.38

結論：

すべての点で「現在のクラスタ重心」が最短距離となりました。

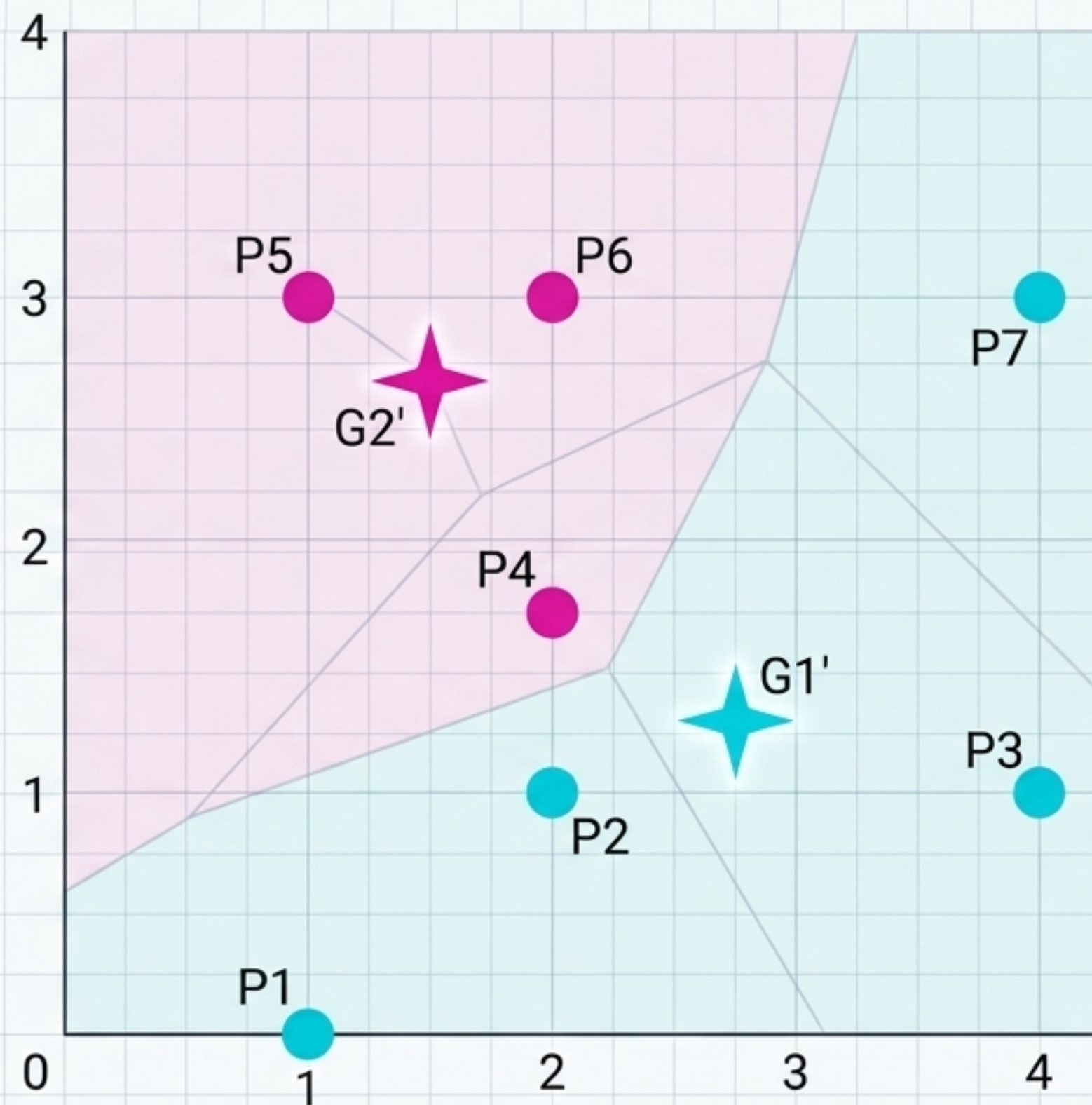
所属の変更（フラグの反転）は発生しないため、ここで計算終了（収束）します。



最終結果：次分類完了（第3回目のループで収束）

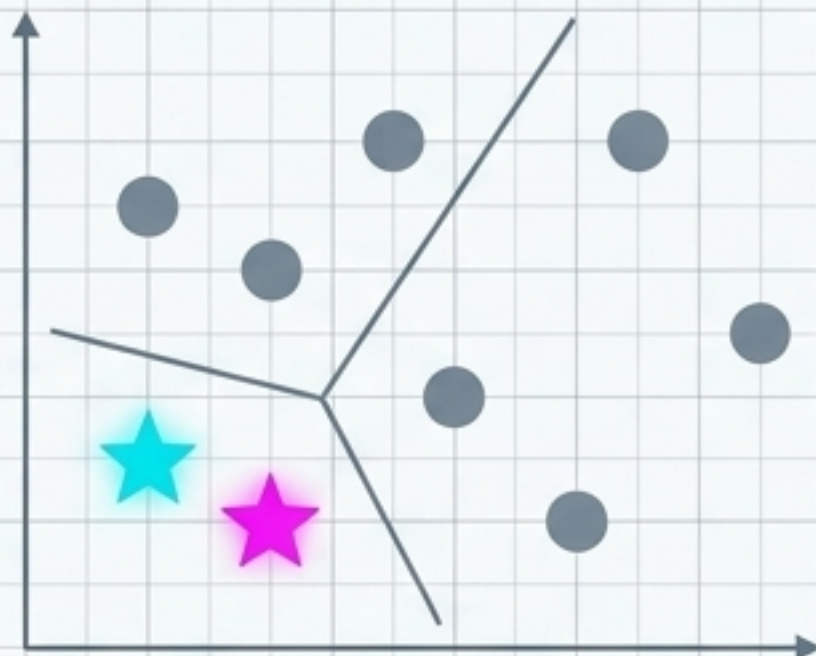
結果は次分類完了：第3回目のループで収束しました。

点 P1: クラスタ 1
点 P2: クラスタ 1
点 P3: クラスタ 1
点 P4: クラスタ 2
点 P5: クラスタ 2
点 P6: クラスタ 2
点 P7: クラスタ 1

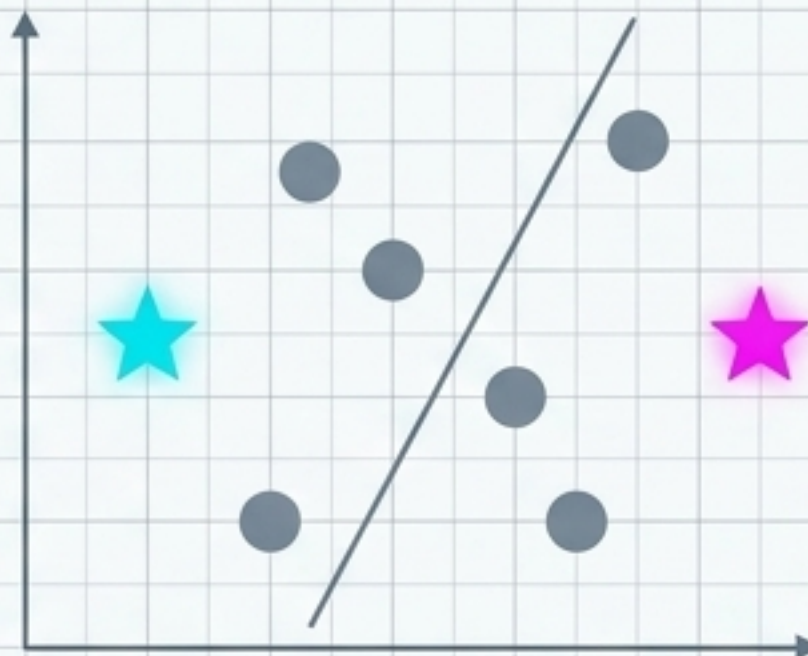


発展：初期値ガチャを防ぐ「K-means++」の思想

⚠ 初期コアが近すぎる失敗例



✓ 距離を離して配置する成功例



弱点：

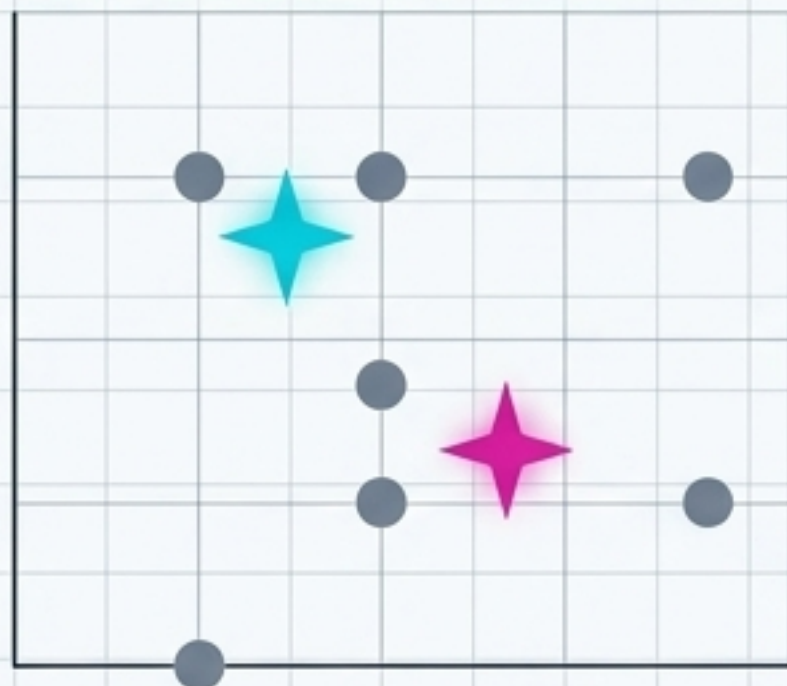
K-meansは最初のコア（初期値）の位置によって、分類結果が最適にならない場合がある。

改良アルゴリズム（試験問題より）：

- 1つ目のコアはランダムに選択。
- 2つ目以降は「すでに決まったコアから離れた点ほど、高い確率で選ばれる」ようにする（距離の和 T_s が大きいほど高確率）。

まとめ：アルゴリズムが描く「最適化の美学」

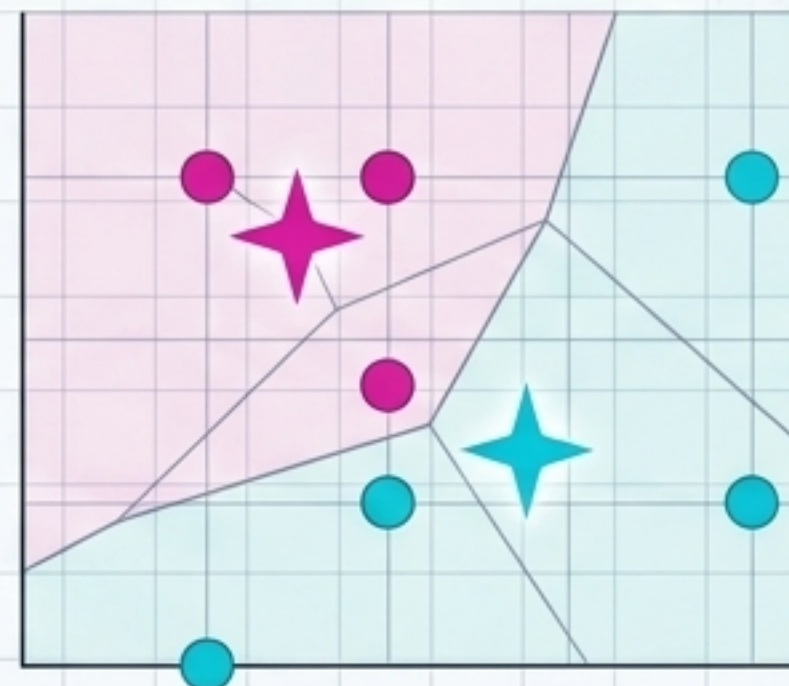
初期状態 (Random)



第1-2ループ (Chaos/Update)



第3ループ収束 (Order)



1. 反復による最適化

完璧な正解を一度の計算で出すのではなく、「評価」と「更新」のループを繰り返すことで正解に到達する。

2. 重心の引力

「個の所属変化」が「集団の重心」を動かし、それがさらに次の「個の所属」を決めるという美しいダイナミズム。

3. 強力な応用範囲

画像処理の減色から、顧客のセグメンテーション、ビッグデータ分析まで、データに潜む構造を暴き出す。