

08

具体例から観るデザイン論

追記型メモリ、Fano 平面、デザイン論入門

めたち (@Metachick_2021)

VRC 数学談話会主催

1|A

追記型メモリの節約術

- 追記型メモリとは？
- クイズ：4回記録の方法
- Fano 平面の使い方

2|A

デザインの定義と例

- Fano 平面の性質
- デザインの定義
- 身近なデザインの例

3|B

デザイン論入門

- デザインの性質
- 発展的な応用

「今日は**組み合わせデザイン論**を解説します！」

追記型メモリの節約術

追記型メモリとは？

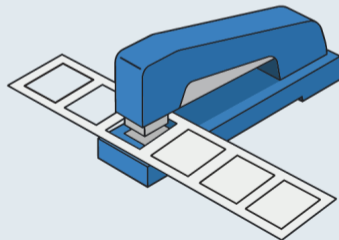
追記型メモリ

以下を満たすようなメモリを**追記型メモリ**という。

- ・ データ（0と1）を記録できるマス目。最初はすべて「0」。
- ・ 一度「1」に書き換えると、絶対に「0」には戻せない。

例えば...

- ・ 紙テープ（パンチカード）
→ 一度開けた穴は塞ぐことができない。
- ・ CD-Rなどのディスク
→ レーザーで焼いた跡は元に戻せない。



追記型メモリとは？

クイズ1

あなたはとある工場のシステム設計担当だ。

工場長：「1~7の整数を1回記録するシステムを作って〜！」

さて、追記型メモリでこれを実現するには、最低何マス必要だろうか？

追記型メモリとは？

クイズ1

あなたはとある工場のシステム設計担当だ。

工場長：「1~7の整数を1回記録するシステムを作って〜！」

さて、追記型メモリでこれを実現するには、最低何マス必要だろうか？

予算がないので、**できるだけ少ないマス目**で作りたい。

追記型メモリとは？

クイズ1

あなたはとある工場のシステム設計担当だ。

工場長：「1~7の整数を1回記録するシステムを作って〜！」

さて、追記型メモリでこれを実現するには、最低何マス必要だろうか？

予算がないので、**できるだけ少ないマス目**で作りたい。例えば、1マスのメモリでは...

- ・ 穴を「開ける」or「開けない」の2通りのパターンが考えられる
- ・ それらを1,2の整数に対応させれば、1~2の整数が表現できる

追記型メモリとは？

同じように考えると...

- ・ 1マスするとき：
前のページで見たように2パターン表現できる。マス目が足りない！
- ・ 2マスするとき：
 $2 \times 2 = 4$ 通りのパターン。まだ足りない。
- ・ 3マスするとき：
 $2 \times 2 \times 2 = 8$ 通りのパターン。これで1~7を表現可能！

追記型メモリとは？

同じように考えると...

- ・ 1マスするとき：
前のページで見たように2パターン表現できる。マス目が足りない！
- ・ 2マスするとき：
 $2 \times 2 = 4$ 通りのパターン。まだ足りない。
- ・ 3マスするとき：
 $2 \times 2 \times 2 = 8$ 通りのパターン。これで1~7を表現可能！

正解

最低 **3マス (3ビット)** 必要！

追記型メモリとは？

以下のように対応させることで無事に工場長の要請をクリアできる
(このアイデアを **2進法** とか、 **2進数** とか呼んだりする)

記録する数字	メモリの状態 (0= 穴なし, 1= 穴あり)	パターン
1	[0, 0, 1]	001
2	[0, 1, 0]	010
⋮	⋮	⋮
7	[1, 1, 1]	111

クイズ2

またまた工場長からのお願い。

工場長：「1~7の整数を4回記録するシステムを作って〜！」

さて、追記型メモリでこれを実現するには、最低何マス必要だろうか？

クイズ2

またまた工場長からのお願い。

工場長：「1~7の整数を4回記録するシステムを作って〜！」

さて、追記型メモリでこれを実現するには、最低何マス必要だろうか？

- ・ 1回目の記録を行った後、1回目の読み取りを行う。
- ・ その後2回目の記録を行い、2回目の読み取りをと繰り返す...

追記型メモリは一度「1」に書き換えると、絶対に「0」には戻せないメモリであったことを思い出しておこう。

アイデア

クイズ1の結果を使えば、12マスで作れることがわかる

アイデア

クイズ1の結果を使えば、12マスで作れることがわかる

- ・ 3マスで1つ目の整数を表すことにして、12マスで4つの整数を作る

アイデア

クイズ1の結果を使えば、12マスで作れることがわかる

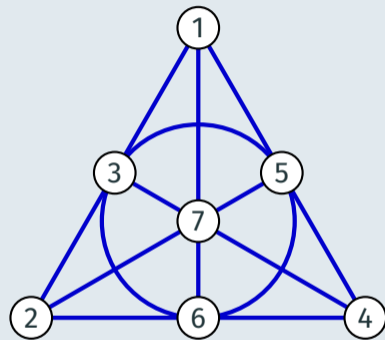
- ・ 3マスで1つ目の整数を表すことにして、12マスで4つの整数を作る



工場長：「12マスは予算が厳しい、助けて！」

→さて、何マスまで減らせるだろうか？

「Fano 平面を導入し、解決する。」



ファノ平面 (PG(2,2))

Fano 平面の使い方 - 1回目

【ルール1：初期状態】

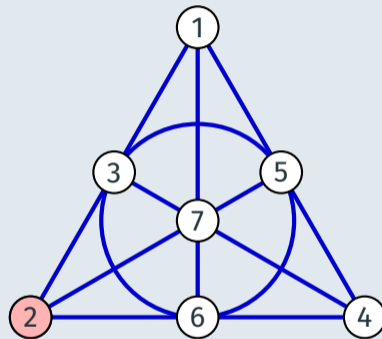
メモリがすべて「0」のとき、数字 i を記録するには位置 i に穴を開ける。（「1」を書き込む）

具体例：数字「2」を記録する

→ 位置2に穴を開ける。

読み取り方

「1」が1つだけのときは、素直にその位置の数字が現在の記録だと解釈する。



Fano 平面の使い方 - 2 回目

【ルール 2 : 1 つ穴がある状態】

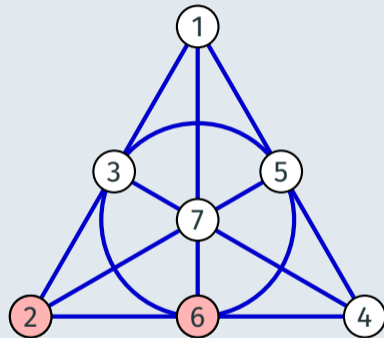
状態 i から新しい数字 j を記録したいときは、図上で i, j, k が直線になるような位置 k に穴を開ける。

具体例 : 状態「2」から数字「4」を記録

→ 2 と 4 を通る直線は $\{2, 6, 4\}$ なので、新しく位置 6 に穴を開ける。

読み取り方

「1」が 2 つのときは、その 2 点を通る直線上の残り 1 点が現在の記録（最新の数字）だと解釈する。



Fano 平面の使い方 - 3回目 - その1

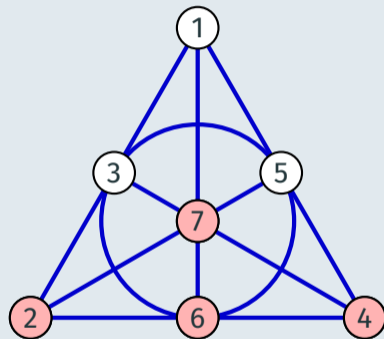
【ルール3：2つ穴がある状態】

数字 l を記録したいときは、新たに2つの穴を開け、
「3点は直線をなし、 l はそこから独立した点」となるようにする。

具体例：状態「2, 6」から数字「7」を記録
→ 4と7に穴を開ける。これで $\{2, 6, 4\}$ が直線となり、
目的の7が独立点になる。

読み取り方

穴が4つのときは、直線（3点）に含まれない独立した1点が現在の記録だと解釈する。



Fano 平面の使い方 - 3 回目 - その 2

【疑問】

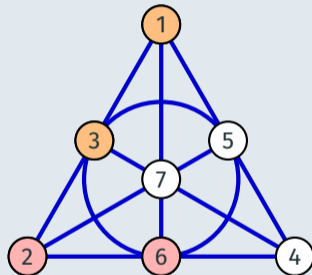
「2」と「6」に穴が開いている（状態 = 4）とき、新しく「6」を記録したい場合はどうする？

【解決策】ルール 3 をそのまま適用する

合計 4 つの穴で「3 点は直線、目的の『6』は独立点」という形を作る。

すでに「2」と「6」があるので、新しく 2 つの穴を追加して、「2」を含む直線を完成させれば OK。

※「7」と「5」を選んでも、読み取る時は「直線以外の点 = 6」となる！



Fano 平面の使い方 - 4 回目 - その 1

【ルール 4 : 4 つ穴がある状態】

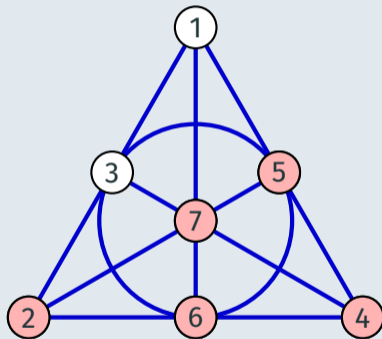
「直線上の数字」を上書き記録したいときは、**独立点とその数字を結ぶ直線上の、空いている位置**に穴を開ける。

具体例 : 状態「2, 6, 4, 7」から「2」を記録

→ 独立点 7 と目的の 2 を結ぶ直線は $\{2, 7, 5\}$ なので、空いている位置 5 に穴を開ける。

読み取り方

穴が 5 つのときは、逆に「穴が開いていない 2 点」を通る直線の、残り 1 点が現在の記録になる。(例 : 空きの 1 と 3 を結ぶ直線 $\{1, 3, 2\}$ の残りである 2)

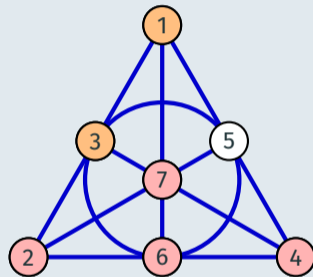


Fano 平面の使い方 - 4 回目 - その 2

【ルール 5：まだ穴がない数字の記録】

4 つ穴がある状態で「まだ穴がない数字」を記録したいときは、**目的の数字「以外」の空いている 2 箇所**に穴を開ける。

具体例：状態「2, 6, 4, 7」から「5」を記録
→ 空いている位置は {1, 3, 5} なので、目的の 5 以外の 1 と 3 に穴を開ける。



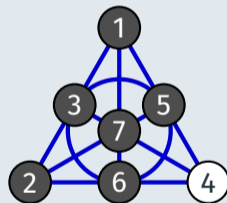
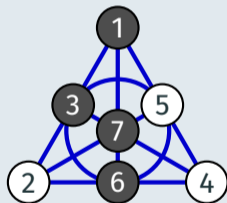
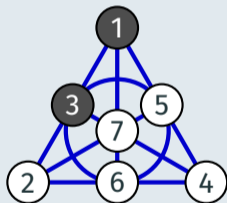
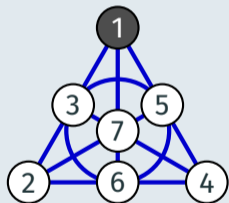
1 と 3 を埋めることで、5 だけが残る

読み取り方

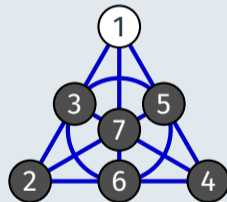
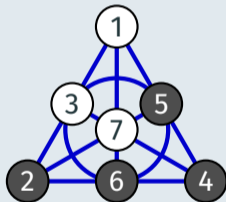
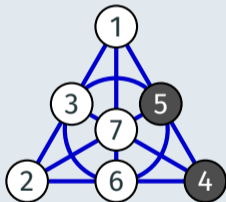
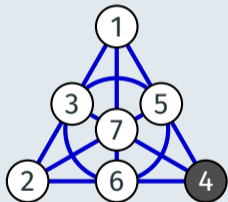
穴が 6 つ（空きが 1 つ）になったときは、「残った 1 つの空き位置」が現在の記録だと解釈する。（例：1 と 3 を埋めて、最後に残った空きの 5）

「実際に読み取ってみる！！」

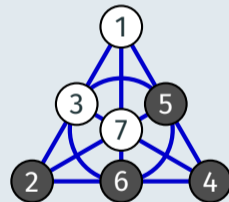
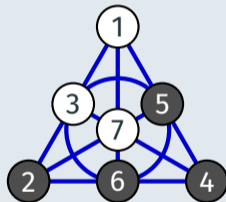
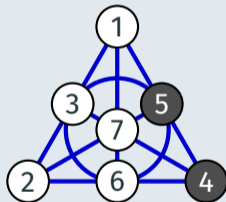
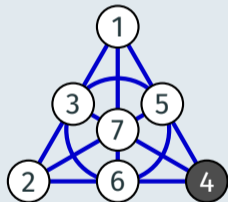
「以下の Fano 平面が表す 4 桁は？」



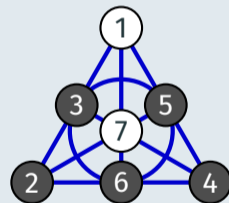
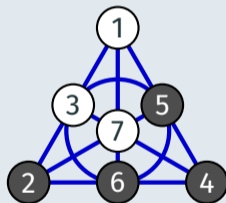
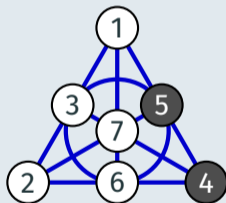
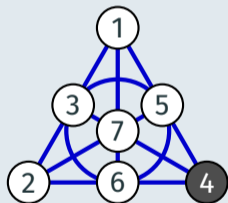
「以下の Fano 平面が表す 4 桁は？」



「以下の Fano 平面が表す 4 桁は？」

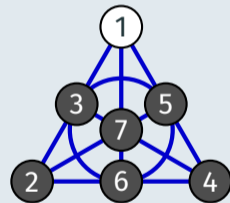
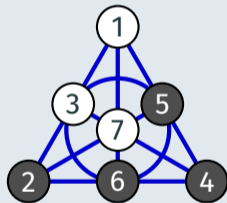
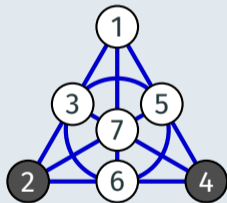
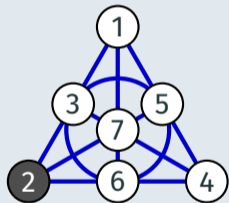


「以下の Fano 平面が表す 4 桁は？」



「2651 を表す Fano 平面は？」

「2651 を表す Fano 平面は？」

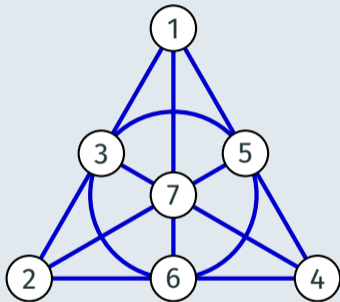


デザインの定義と例

「なぜ Fano 平面を使うとうまくいった？」

「とてつもない対称性」を持つから！

- ・ 7つの点の中から
- ・ どの2つ点を選んでも
- ・ ちょうど1本の直線に含まれる
- ・ さらに、どの直線もちょうど3つの点を含む



「このような対称的な構造をデザインという！」

定義1 デザインの定義

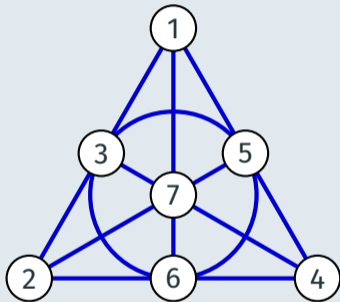
$t, v, k, \lambda \in \mathbb{N}$ が $v \geq k \geq t$ を満たすとする。 $t - (v, k, \lambda)$ デザインとは、集合 \mathcal{P} と \mathcal{P} の部分集合族 \mathcal{B} の組 $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ であって、次の条件を満たすもののことである。

- (1) $|\mathcal{P}| = v$ である。
- (2) \mathcal{B} の各要素は \mathcal{P} の k -要素部分集合である。
- (3) \mathcal{P} の任意の t -要素部分集合は \mathcal{B} のちょうど λ 個の要素に含まれる。

\mathcal{D} がデザインである時、 \mathcal{P} の要素を点、 \mathcal{B} の要素をブロックと呼ぶ。

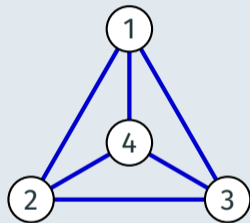
Fano 平面は「2-(7,3,1) デザイン」である！

- (1) 7つの点がある ($v = 7$)
- (2) 各直線（ブロック）はちょうど3つの点を含む ($k = 3$)
- (3) どの2つの点を選んでも、ちょうど1本の直線に含まれる ($t = 2, \lambda = 1$)



完全グラフ K_4 は「 $2-(4,2,1)$ デザイン」！

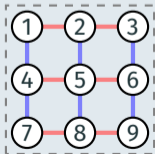
- (1) 4つの点がある ($v = 4$)
- (2) 各辺 (ブロック) はちょうど2つの点を含む ($k = 2$)
- (3) どの2つの点を選んでも、ちょうど1本の辺に含まれる ($t = 2, \lambda = 1$)



有限アフィン平面 $AG(2, 3)$ は「2-(9,3,1) デザイン」！

- (1) 9つの点がある ($v = 9$)
- (2) 各直線（ブロック）はちょうど3つの点を含む ($k = 3$)
- (3) どの2つの点を選んでも、ちょうど1本の直線に含まれる ($t = 2, \lambda = 1$)

【縦・横】



【斜め（右下がり）】



【斜め（右上がり）】



身近なデザインの例

大ヒットゲーム「ダブル (Dobble)」のルール

8種類の図形が描かれた丸いカードがあり、これらはどの2枚の組み合わせでも同じ柄の図形が必ず1組だけある。

- ・各プレイヤーは、**山札と自分のカードの共通の図形**をいち早く探して取る。
 - ・これを繰り返し、最終的にカードをたくさん獲得した人が勝ち！
-
- ・子供から大人まで、直感的に遊べて盛り上がるスピード勝負のゲーム。
 - ・ルール説明：
https://www.youtube.com/watch?v=trZhC1qXC_k



ドブルの数学的背景

ドブルの原理は $2-(57, 8, 1)$ デザインそのもの！

- ・ $v = 57$: 登場する図形は全部で 57 種類 (点)。
- ・ $k = 8$: 1 枚のカードには 8 つの図形が描かれている (ブロック)。
- ・ $\lambda = 1$: どの 2 つの図形を選んでも、ちょうど 1 枚のカードで同時に出現する。
- ・ **双対性** : その 2 つのカードを選んでも、ちょうど 1 つの図形が同時に出現する

数学史に残るパズル「カークマンの女学生問題」

15人の女学生が、毎日3人ずつ5つのグループに分かれて散歩に出かける。「どの2人の女学生も、必ず1回だけ同じグループになる」ように、7日間のスケジュールを組めるだろうか？

- ・ 1850年にイギリスの数学者トーマス・カークマンが雑誌に投稿した有名な問題。
- ・ ルールはシンプルだが、手作業で条件を満たすようにスケジュールを組もうとすると、すぐに矛盾が生じてしまう難問。

女学生問題の数学的背景

この問題は、15個の要素からなる $2-(15, 3, 1)$ デザインを構成するだけでなく、それを曜日ごとに綺麗に分割するという、さらに良い性質を求めている。

- $v = 15$: 15人の女学生 (点)。
- $k = 3$: 3人ずつのグループ (ブロック)。
- $\lambda = 1$: どの2人も必ず1回だけ同じグループに。
- **分解可能性** : 作られた全35個のグループを、「15人全員が網羅される5個ずつのまとめり (= 1日分のスケジュール)」へと綺麗に7分割できるかが、この問題の最大のポイントである。

農事試験場での課題：「公平な肥料テスト」

複数の新しい肥料（または品種）の効果を比較したい。しかし、畑は場所によって日当たりや水はけなどの条件がバラバラである。「特定の肥料だけ条件の良い場所にまかれた」という不公平をなくし、正確な比較データを得るにはどう配置すべきだろうか？

- ・ 1920年代、天才統計学者 R.A. フィッシャーがこの問題を解決するために考案したのが「実験計画法」である。
- ・ 単なる数学のパズルだったデザイン理論が、農業や科学実験、さらには現代の Web マーケティング（A/B テストなど）を支える実用的なツールへと進化した瞬間である。

肥料の配置を「ブロックデザイン」で解決

条件の似た畑の区画をグループ（ブロック）に分け、そこにバランスよく肥料（点）を配置する。「どの2つの肥料の組み合わせも、同じ条件で必ず指定された回数だけ一緒にテストされる」ように計画を組む。

- ・ v (Varieties) : 比較したい肥料や品種の数（デザインの「点」）。
- ・ b (Blocks) : 畑のまとまった区画（デザインの「ブロック」）。
- ・ r (Replications) : 各肥料をテストする反復回数。
- ・ **記号の由来** : 現在でもデザイン理論で v や b, r という文字が使われているのは、この農業実験の用語（品種 = Varieties、反復 = Replication）の名残である。

デザイン論入門

デザイン理論の最大の謎：「そもそも存在するのか？」

パラメータ (v, k, λ など) が計算上の条件を綺麗に満たしていても、実際にその配置を構成できる、つまり「存在する」とは限らない。

- ・ 一般に、与えられた、 t, v, k, λ に対してそれを満たすデザインが存在するかは未解決
- ・ デザイン論の大きな研究の方法の一つは、「デザインが存在するための必要条件」を与えること

デザインの性質

定義. シュタイナー3重系 (存在性が完全にわかっている例)

ブロックのサイズがすべて3 ($k=3$) で、どの2点も必ず1回だけ同じブロックに入る ($\lambda=1$) ような $2-(v,3,1)$ デザインを「シュタイナー3重系 (STS)」と呼ぶ。

定理. カークマンの定理 (1847年)

STS が存在するための必要十分条件は、点の数 v を6で割った余りが1または3になることである。

- ・ Fano 平面や有限アフィン平面 $AG(2,3)$ 、女学生問題はすべて STS である。
- ・ 存在は証明されていても、STS の個数は未解明である。 $v=15$ の STS は 80 種類だが、 $v=19$ になると一気に 110 億種類以上になることが知られている。

定義・命題. 補デザイン (Complementary Design)

点集合 V とブロック集合 \mathcal{B} からなる $2 - (v, k, \lambda)$ デザイン $D = (V, \mathcal{B})$ を考える。このとき、各ブロック $B \in \mathcal{B}$ の補集合 $\bar{B} = V \setminus B$ を新たなブロックとする構造 $\bar{D} = (V, \bar{\mathcal{B}})$ もまたデザインとなり、これを補デザインと呼ぶ。

- ・ 元のデザインのブロック数を b 、反復数を r としたとき、補デザイン \bar{D} は $2 - (v, v - k, b - 2r + \lambda)$ デザインとなる。
- ・ 複雑な代数的計算を要さず、補集合をとるという基礎的な集合操作のみで、未知のパラメータを持つデザインの存在を自動的に導ける強力な命題である。

デザイン論を支える数学的手法

デザインのパラメータが満たすべき条件やその存在性を調べるために、組合せ数学や代数学の強力な手法が応用される。

- ・ ダブルカウントによる基本関係式：
なんやかんや**組み合わせ論的な議論**によって、 $vr = bk$ や $r(k - 1) = \lambda(v - 1)$ といった関係式が導かれる。
- ・ 結合行列と線形代数学の導入：
点とブロックの配置を「0と1を並べた行列（結合行列）」として表現することで、行列式や固有値といった**線形代数**の理論を応用し、デザインが存在するかどうかを代数的に判定できる。

デジタル社会を支える「符号理論」との交差点

QRコードや宇宙通信などで、データ通信のノイズ（誤り）を自動的に修復する**誤り訂正符号**の構成にデザインが応用されている。

- ・ デザインの点とブロックの関係を表す「0と1の行列」をうまく使うことで、効率的で復元能力の高いデジタル暗号や符号を作ることができる。
- ・ ボイジャー探査機の画像通信などにも使われた**ゴレイ符号**という非常に優秀な符号の背後には、美しいパラメータを持つデザインが隠れている。

有限群論の至宝「散在型単純群」との繋がり

デザインが持つ図形としての対称性の美しさを突き詰めると、純粋代数学における群論の深い謎に行き着く。

- ・ あるデザインの構造を保ったまま点やブロックを入れ替える操作（自己同型）を集めると、群と呼ばれる代数的な構造になる。
- ・ 群の世界に例外的にしか存在しない「散在型単純群」と呼ばれる群が、特定の美しいデザインの自己同型群として自然な形で現れることが知られている。

最先端の物理学への拡張「量子的 t-デザイン」

古典的な組合せ数学の枠組みであったデザイン論の考え方が、近年量子力学の世界（ヒルベルト空間）へと拡張されている。

- ・ 無限に存在する複雑な量子状態の中から、全体をうまく近似できる代表的な有限個のセットを選ぶ数学的な手法として用いられる。
- ・ 量子暗号の安全性証明や、量子コンピュータのアルゴリズムの効率化など、現在進行形で活発に研究されている最先端のトピックである。たぶん。

- J. H. van Lint and R. M. Wilson: *A Course in Combinatorics, Second Edition*.
- Mohammad Derakhshani: *Quantum t -design*, Master's thesis, University of Waterloo, 2008.