



Tokyo Tech

離散構造とアルゴリズム (4-1) アルゴリズムの設計技法

高橋篤司

東京工業大学 工学院 情報通信系

アルゴリズムの設計

■ 出力による分類

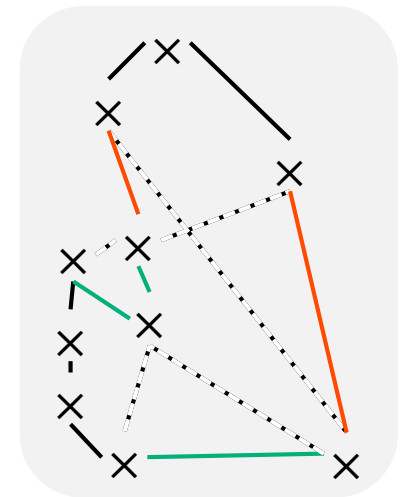
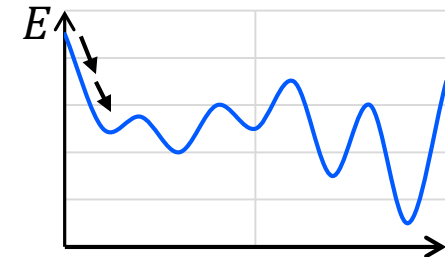
- (厳密)アルゴリズム(exact)
 - 最適解, 厳密解
- 近似アルゴリズム(approximate)
 - 近似解 ← 近似比(厳密解の質との比)が既知
- 発見的アルゴリズム(heuristic)

■ 方法による分類

- 構成的アルゴリズム(constructive)
 - 部分解から解
- 反復的(探索的)アルゴリズム(iterative)
 - 初期解を修正
 - 勾配法(gradient)
 - 確率的アルゴリズム(stochastic)

■ 代表的な設計技法

- 分枝限定(branch and bound)
- 分割統治(divide and conquer)
- 動的計画法(dynamic programming)
- 貪欲アルゴリズム(greedy)



平面の10点を巡回せよ

確率的探索アルゴリズム (stochastic search)

- 焼きなまし法(SA, simulated annealing)
 - 物質のエネルギー状態
- 遺伝的アルゴリズム(GA, genetic algorithm)
 - 生命の進化
- ...
- ランダム探索(random search)

- ✓ 自然界の振る舞いから着想を得たものが多い
 - 構造や変数の選択・調整がポイント
 - hyper-parameterの選択
 - 調整方法(parameter tuning(学習learning)) の選択
 - 自動化は永遠の課題?
 - すべてに対応する必要なし, 必要な場面で役立つためには?

シミュレーテッド・アニーリング (SA)

■ 焼きなまし法(SA, simulated annealing)

- 熱い鉄をたたくと強い鉄ができる
 - ひずみの解消, 低内部エネルギー状態に向かう

■ 局所修正 (隣接解への遷移) の繰り返し

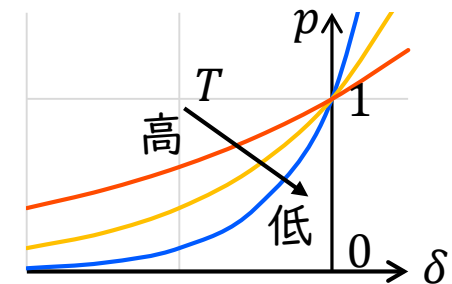
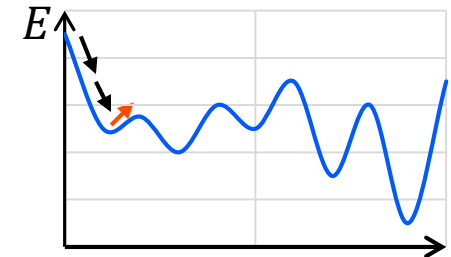
- 遷移確率 $p = \exp\left(\frac{\delta}{T}\right)$: (δ :ゲイン, T :温度)
 - 改善($\delta \geq 0$) $\Rightarrow p = 1$ (必ず遷移)
 - 改悪($\delta < 0$) $\Rightarrow 0 < p < 1$ (改悪量小, 温度大で遷移確率大)
 - 局所最適からの脱出

■ 温度スケジュール

- 平衡状態を維持しながら高温から低温へ
 - 高温域 : 大域的探索
 - 低温域 : 局所的探索 (ほぼ山登り法, hill climbing)
- ✓ 適切な温度スケジュール : 最適解が得られる確率を1に近づけられる
 - ただし時間がかかる

■ 解空間 (点:解, 辺:隣接) の定義

- 直径と次数 : 最適解への遷移回数は小さいが, 隣接解が多すぎないのがよい
- なめらかさ : 近傍で評価の変化が小さい場合のがよい



遺伝的アルゴリズム (GA)

■ 遺伝的アルゴリズム (genetic algorithm, GA)

- 個体の染色体(chromosome)表現

■ 遺伝子(gene)の選択

- 集団形成と進化

■ 次世代の生成

a. 交叉(crossover)

■ 優秀な親から子を生成

■ 致死遺伝子(非許容解)の回避

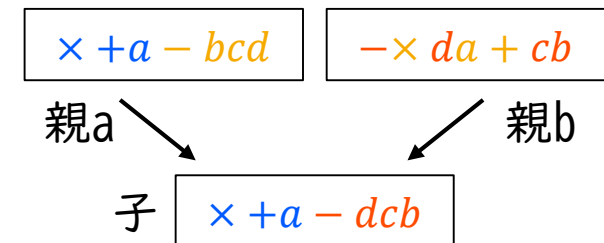
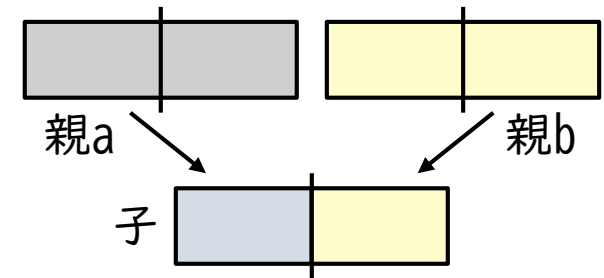
b. 突然変異(mutation)

■ 多様性の維持(集団の進化には必須)

c. 選択(selection)

■ 個体の評価(fitness function)

■ 多様性を維持しつつ,よい解を多く選択する



ニューラルネットワーク (NN)

■ ニューラルネットワーク(neural network)

- 脳の挙動を模倣

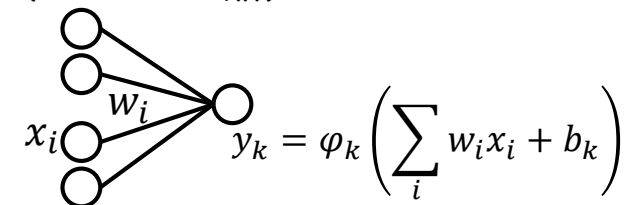
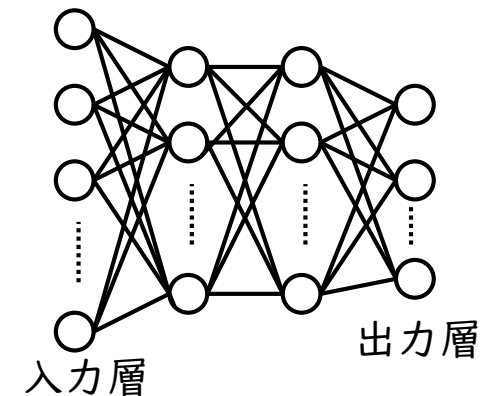
- ニューロン(neuron)の反応が伝搬
- 刺激の履歴で結合強度が変化

- ネットワーク構造

- 入力層・隠れ層(直接の観察制御不能)・出力層
- 層間結合
 - 全結合, 畳み込み(近傍相関抽出), プーリング(データ圧縮)

- パラメータ調整法

- 誤差伝搬法(backpropagation)
 - 多数層の場合に難点
- 多数層(deep layer)に対する調整方法(deep-learning)の発展



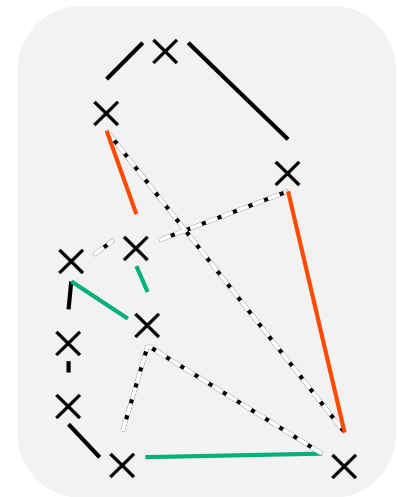
分枝限定(branch and bound)

➤ 全探索の効率化

1. 分枝: 解を詳細化(場合分け, 解集合の再帰的分割)
2. 限定(枝刈り): 不要な分枝(詳細化)を打ち切る
3. 選択: 残った解候補から最良解を選択

■ 解集合(分枝)の評価

- 評価値の上下限を利用
- 効率よくタイトに評価できるか?
 - 厳密な評価 ⇒ 厳密解(指数時間アルゴリズム?)
 - 発見的評価 ⇒ 解(厳密解ではない)
 - 機械学習の利用?



平面の10点を巡回せよ

分割統治(divide and conquer, and merge)

➤ 小さいと治めやすい(大きいと厄介)

1. 分割: 同じ種類の部分問題に(再帰的に, いくつに? 大きさは?)
2. 統治: 部分解を得る(簡単な問題になったら)
3. 併合: 部分解から全体の解を得る(併合の手間は?)

■ 例

- 併合整列アルゴリズム

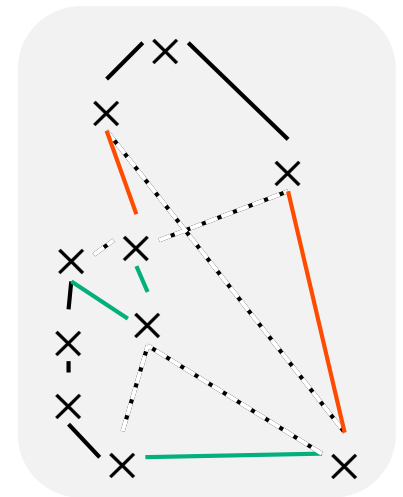
■ 大きさが半分の二つの部分問題に分割

✓ $T(n) = 2T\left(\frac{n}{2}\right) + O(n) = O(n \log n)$

- ハノイの塔

■ 大きさが1小さい二つの部分問題に分割

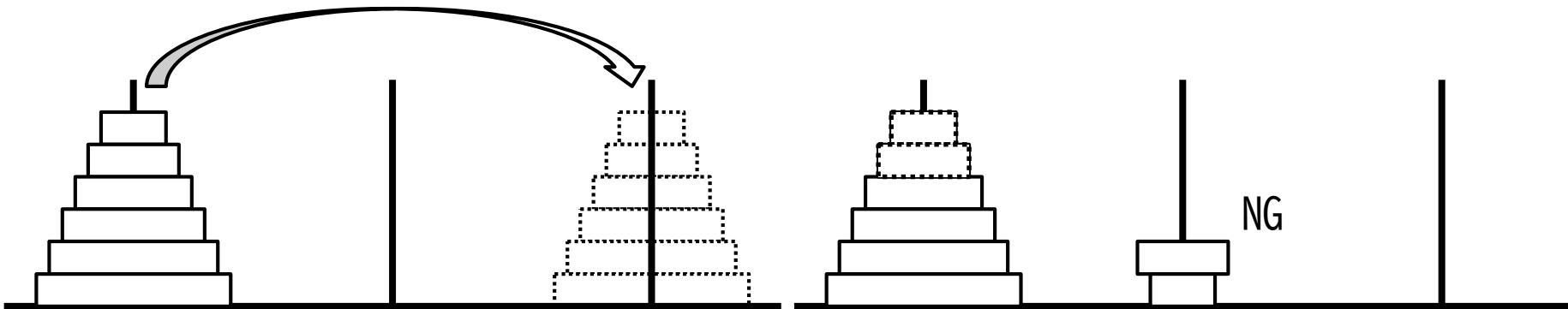
✓ $T(n) = 2T(n-1) + O(1) = O(2^n)$



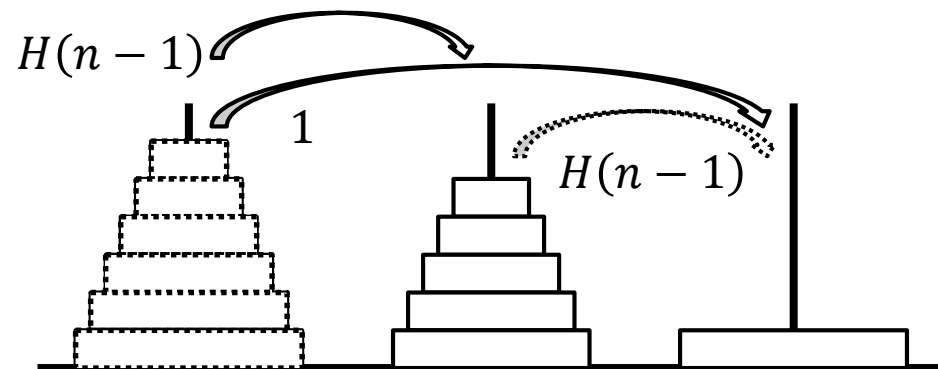
平面の10点を巡回せよ

ハノイの塔

- 3本の杭の1本に, 円盤が大きい順に積み重ねられている
- すべての円盤を別の1本の杭に移動させよ
 - 円盤を1枚ずつ別の杭に移動させる
 - ただし, 小さい円盤の上に大きい円盤を置いてはいけない

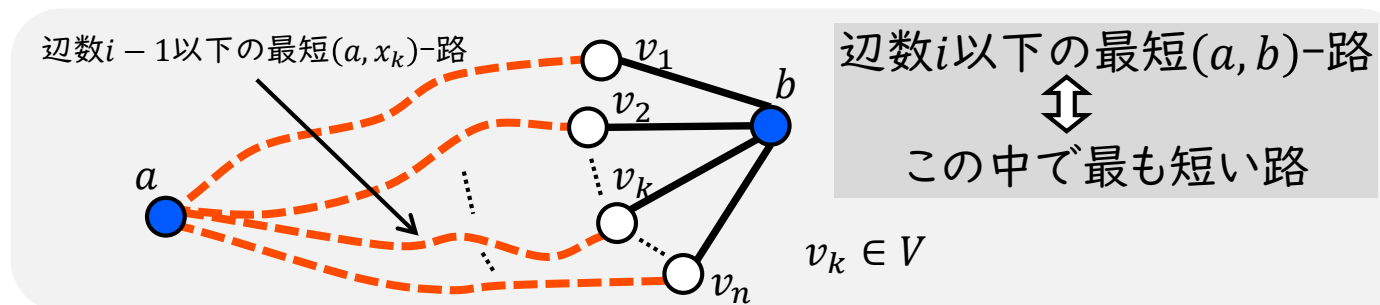


- 移動の手順は?
- 移動回数 $H(n)$ は?
 - 漸化式
 - $H(n) = 2H(n-1) + 1$
 - $H(1) = 1$
 - $H(n) = 2^n - 1$



動的計画法(dynamic programming)

- 部分問題とその解を小さい順に生成する
- 最適性の原理
 - 問題の最適解は, 部分問題の最適解から構成される
が成立する場合
- 例
 - 最短路問題
 - 最短路
 - 辺数 i 以下の最短(a, b)-路
 - ⇒ 辺数 $i - 1$ 本以下の最短(a, v_k)-路+辺(v_k, b)



動的計画法(dynamic programming)

➤ 部分問題とその解を小さい順に生成する

■ 最適性の原理

- 問題の最適解は, 部分問題の最適解から構成される
が成立する場合

■ 例

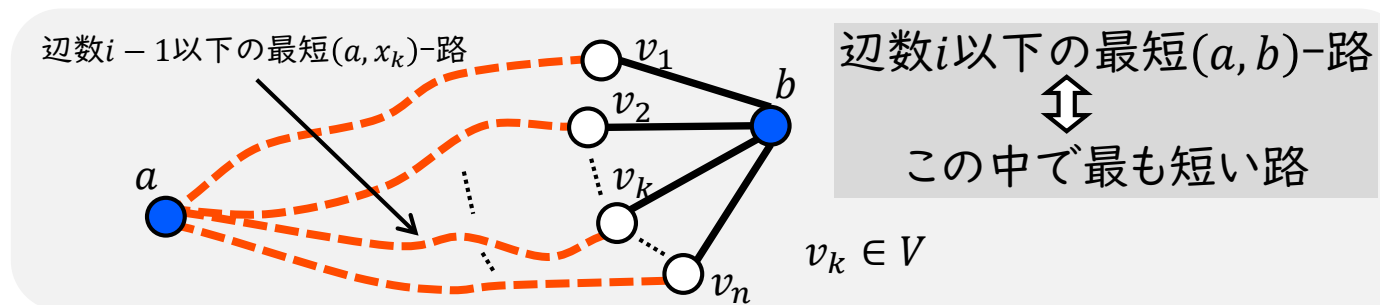
- 最短路問題

■ Bellman-Ford

- $\forall v \in V$, 辺数 i 以下の最短(a, v)-路を構成 (i の値を順に増加させる)
 - 辺数 $i - 1$ 本以下の最短(a, v_k)-路+辺(v_k, v)から

■ Dijkstra

- 距離確定点 v_k までの最短(a, v_k)-路+辺(v_k, v)を構成



貪欲アルゴリズム(greedy algorithm)

➤ 目先の利益を追求する

- 各時点で評価が最大になるように要素の追加を繰り返す
- 許容解(条件を満たす解)を膨らませていく

✓ 厳密解が得られるとは限らない

■ 例

- 最大全域木

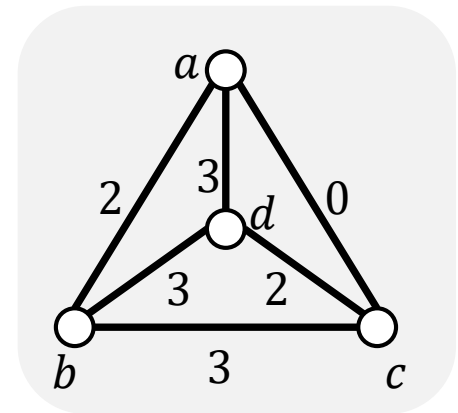
■ Kruskalアルゴリズム(厳密)

■ Primアルゴリズム(厳密)

- 最大巡回セールスマン問題

■ 入力: 完全グラフ K_n , 重み関数 $w: E \rightarrow \mathbb{R}^+$

■ 質問: 重み最大のハミルトン閉路を求めよ



独立系 (independence system)

■ 性質 P

- 集合 E の部分集合に対して定義
- $I = \{X \subseteq E \mid X \in P\}$

■ 独立系 $M = (E, I)$

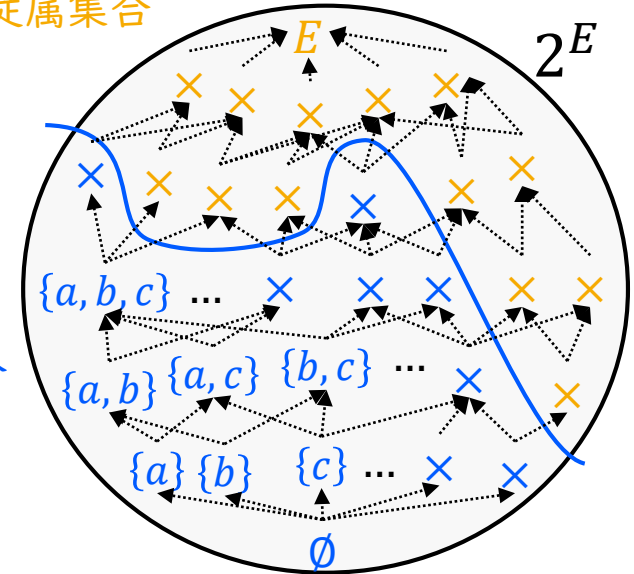
1. $\emptyset \in I \quad \therefore$ 空集合は P を満たす
2. $\forall X \in I, Y \subseteq X \Rightarrow Y \in I$
 $\therefore P$ を満たす集合の部分集合は P を満たす

独立系

$$M = (E, I)$$

独立集合

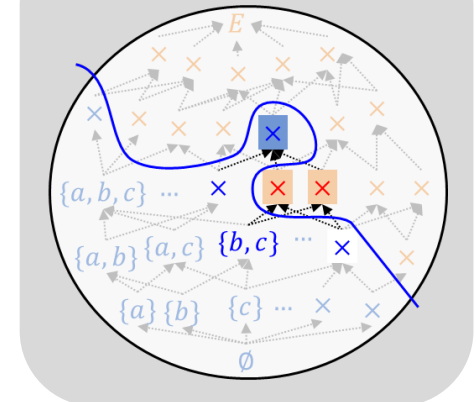
従属集合



■ 独立集合 (independent set): P を満たす集合

■ 従属集合 (dependent set) : P を満たさない集合

独立系ではない



独立系 (independence system)

■ 性質 P

- 集合 E の部分集合に対して定義
- $I = \{X \subseteq E \mid X \in P\}$

■ 独立系 $M = (E, I)$

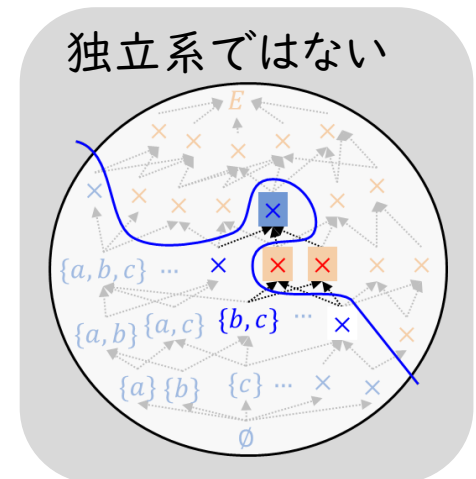
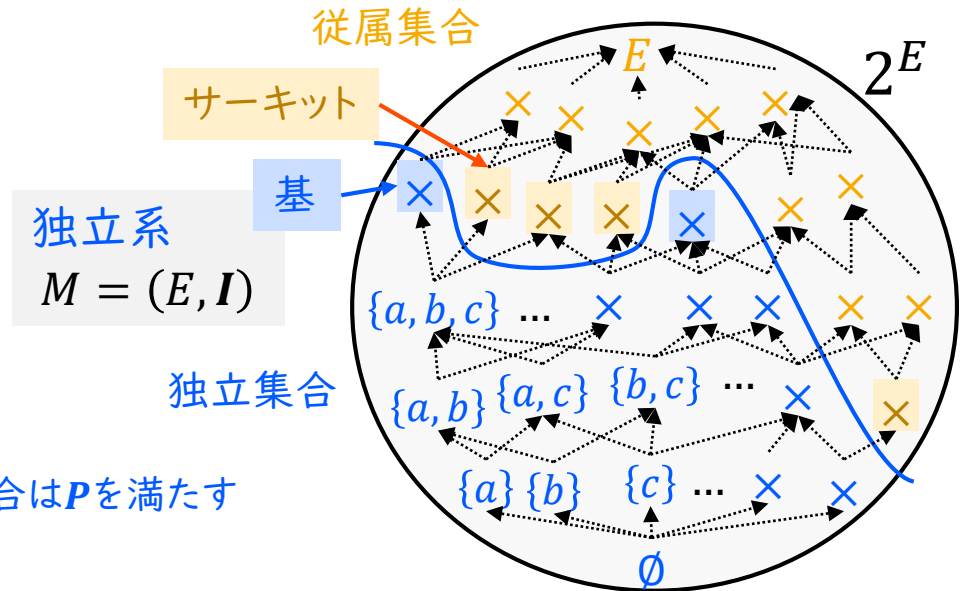
1. $\emptyset \in I \quad \therefore$ 空集合は P を満たす
2. $\forall X \in I, Y \subseteq X \Rightarrow Y \in I$
 $\therefore P$ を満たす集合の部分集合は P を満たす

■ 独立集合 (independent set): P を満たす集合

- 基 (base) : 極大な独立集合

■ 従属集合 (dependent set) : P を満たさない集合

- サーキット (circuit) : 極小な従属集合



独立系 (independence system)

■ 性質 P

- 集合 E の部分集合に対して定義
- $I = \{X \subseteq E \mid X \in P\}$

■ 独立系 $M = (E, I)$

1. $\emptyset \in I \quad \therefore$ 空集合は P を満たす
2. $\forall X \in I, Y \subseteq X \Rightarrow Y \in I$
 $\therefore P$ を満たす集合の部分集合は P を満たす

■ 貪欲アルゴリズム

- 入力 : $M = (E, I)$, 重み関数 $w: E \rightarrow \mathbb{R}$

 1. $\emptyset \in I \quad \therefore$ 初期解あり (解を出力)
 2. $\forall X \in I, Y \subseteq X \Rightarrow Y \in I$
 $\therefore$ 性質を満たす任意の解に到達可 (基を出力)

