大阪電気通信大学 情報通信工学部 光システム工学科 2年次配当科目

コンピュータアルゴリズム

様々な技法

第11講: 平成21年1月9日 (金) 4限 E252教室

中村 嘉隆(なかむら よしたか)
奈良先端科学技術大学院大学 助教
y-nakamr@is.naist.jp
http://narayama.naist.jp/~y-nakamr/

第 12 講の復習

- グラフアルゴリズム
  - 最短路の問題
    - ダイクストラのアルゴリズム
  - 最小木の問題
    - プリムのアルゴリズム

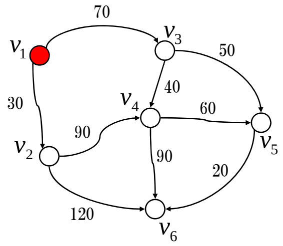
最短路を求める問題

- ダイクストラ(Dijkstra)のアルゴリズム
  - ある頂点 s から他の各頂点への最短経路を求めるための効率の良いアルゴリズム
  - 概要
    - 重み(ただし正の数)のつけられたグラフにおいて最短経路を求めるアルゴリズム
    - 対象はすべての頂点が連結されたグラフとする
  - 基本戦略
    - 各頂点の最短経路を出発点に近い(最短経路の長さが短い)ものから一つずつ確定していく
  - 性質
    - 最短経路が(経路があれば)必ず見つかることが保証されている

復習：ダイクストラ法

- 最短経路の表を完成させよ

f1	0	–
f2	∞	–
f3	∞	–
f4	∞	–
f5	∞	–
f6	∞	–



P:最短距離計算済頂点

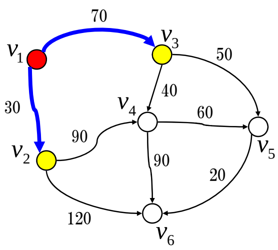
T:次候補頂点

v_1

復習：ダイクストラ法 –ステップ 1–

- 最短経路の表を完成させよ

f1	0	–
f2	30	1
f3	70	1
f4	∞	–
f5	∞	–
f6	∞	–



P:最短距離計算済頂点

v_1

T:次候補頂点

v_2 v_3

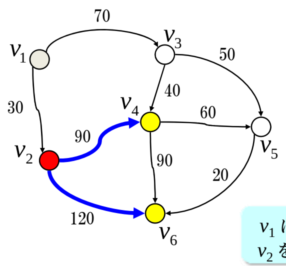
復習：ダイクストラ法 –ステップ 2–

- 最短経路の表を完成させよ

f1	0	–
f2	30	1
f3	70	1
f4	120	2
f5	∞	–
f6	150	2

30+90

30+120



v_1 に最も近い
v_2 を次は基準

P:最短距離計算済頂点

v_1 v_2

T:次候補頂点

v_3 v_4 v_6

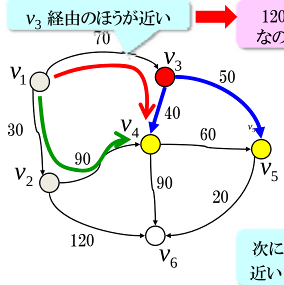
復習：ダイクストラ法 －ステップ 3－

- 最短経路の表を完成させよ

v_3 経由のほうが近い

120 > 70+40
なので上書き

70+50



f1	0	-
f2	30	1
f3	70	1
f4	110	3
f5	120	3
f6	150	2

P:最短距離計算済頂点

v_1 v_2 v_3

次に v_1 に最も近い v_3 を基準

T:次候補頂点

v_4 v_6 v_5

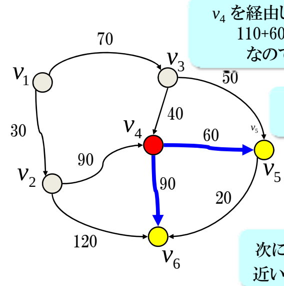
2009/1/9 第11講 様々な技法 7

復習：ダイクストラ法 －ステップ 4－

- 最短経路の表を完成させよ

v_4 を経由して v_5 に行くと
110+60=170 > 120
なのでそのまま

v_6 も同様
110+90 > 150



f1	0	-
f2	30	1
f3	70	1
f4	110	3
f5	120	3
f6	150	2

P:最短距離計算済頂点

v_1 v_2 v_3 v_4

次に v_1 に最も近い v_4 を基準

T:次候補頂点

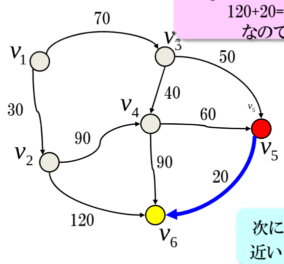
v_6 v_5

2009/1/9 第11講 様々な技法 8

復習：ダイクストラ法 －ステップ 5－

- 最短経路の表を完成させよ

v_5 を経由して v_6 に行くと
120+20=140 < 150
なので上書き



f1	0	-
f2	30	1
f3	70	1
f4	110	3
f5	120	3
f6	140	5

P:最短距離計算済頂点

v_1 v_2 v_3 v_4 v_5

次に v_1 に最も近い v_5 を基準

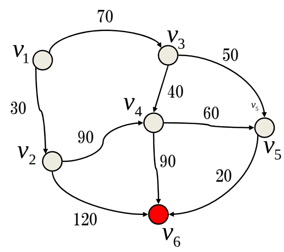
T:次候補頂点

v_6

2009/1/9 第11講 様々な技法 9

復習：ダイクストラ法 －終了－

- 最短経路の表を完成させよ



f1	0	-
f2	30	1
f3	70	1
f4	110	3
f5	120	3
f6	140	5

P:最短距離計算済頂点

v_1 v_2 v_3 v_4 v_5 v_6

T:次候補頂点

2009/1/9 第11講 様々な技法 10

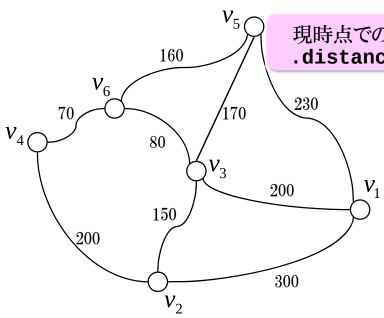
最小木を求める問題

- 最小木（MST: Minimum Spanning Tree）
  - すべての頂点を連結する木で，辺の重みの総和が最小のもの
- プリム（Prim）のアルゴリズム
  - ダイクストラ法と基本は同じ
  - 基本戦略
    - 最も重みの小さな辺から順に最小木の枝になるかどうか調べていく

2009/1/9 第11講 様々な技法 Page 11

プリム法 － 初期状態 －

現時点での .distance



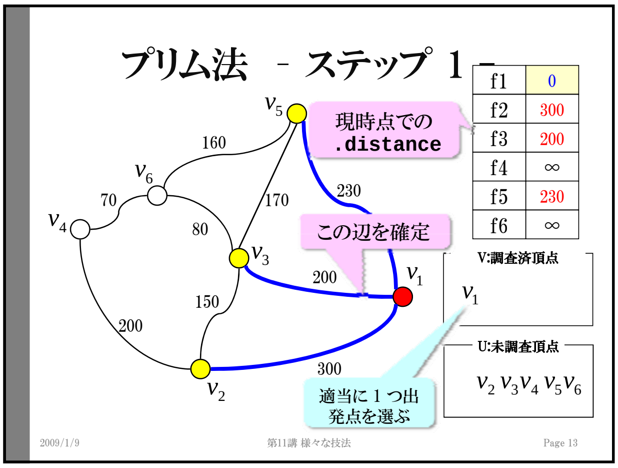
f1	∞
f2	∞
f3	∞
f4	∞
f5	∞
f6	∞

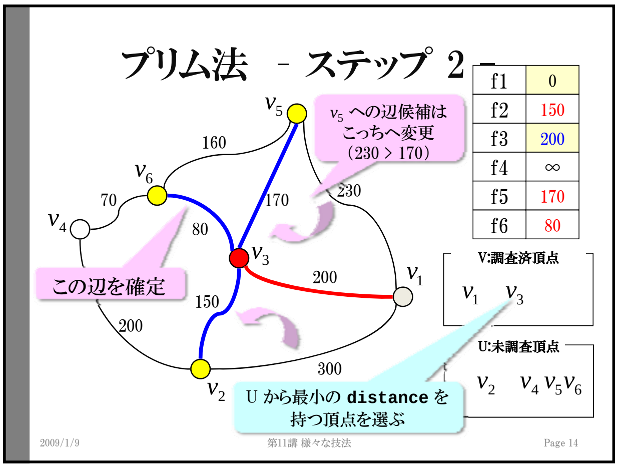
V:調査済頂点

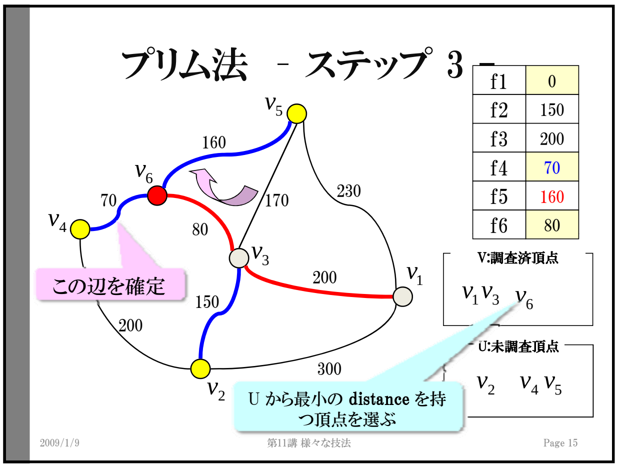
U:未調査頂点

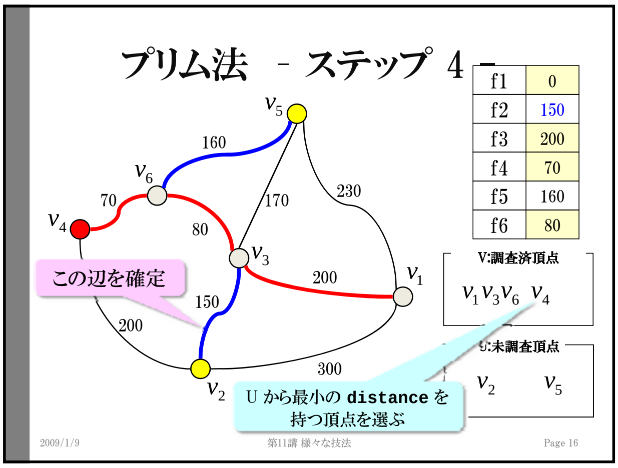
v_1 v_2 v_3 v_4 v_5 v_6

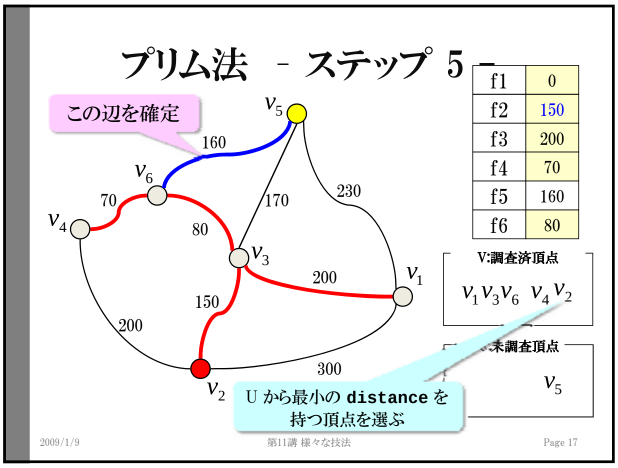
2009/1/9 第11講 様々な技法 Page 12

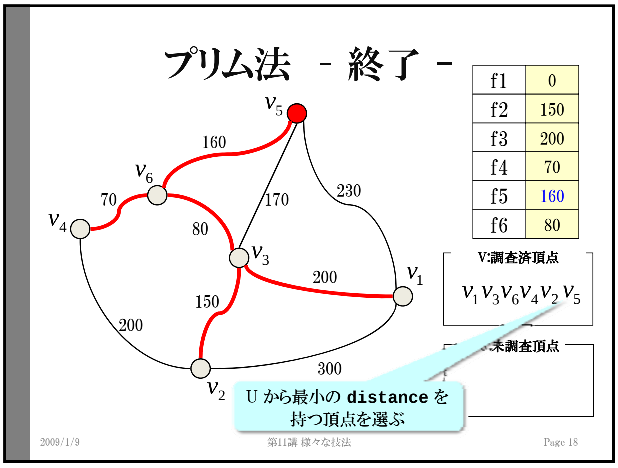

プリム法 - ステップ 1 -
現時点での .distance
この辺を確定
適当に 1 つ出発点を選ぶ
160
70
80
170
230
200
150
200
300
f1 0
f2 300
f3 200
f4 ∞
f5 230
f6 ∞
V:調査済頂点
v_1
U:未調査頂点
$v_2\ v_3 v_4\ v_5 v_6$
2009/1/9
第11講 様々な技法
Page 13

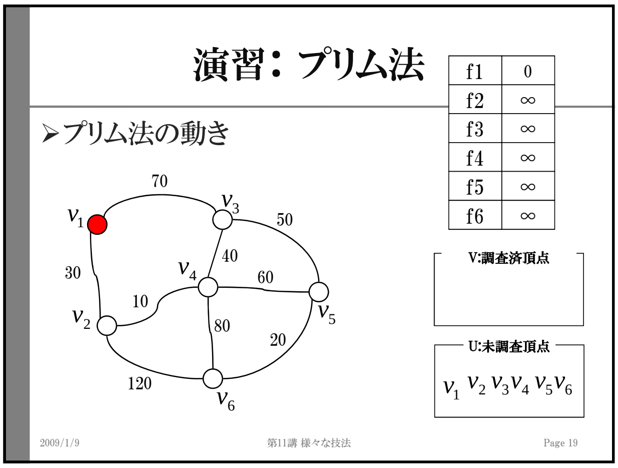

プリム法 - ステップ 2 -
v_5 への辺候補は こっちへ変更 (230 > 170)
この辺を確定
U から最小の distance を 持つ頂点を選ぶ
f1 0
f2 150
f3 200
f4 ∞
f5 170
f6 80
V:調査済頂点
$v_1\ \ v_3$
U:未調査頂点
$v_2\ \ v_4\ v_5 v_6$
2009/1/9
第11講 様々な技法
Page 14

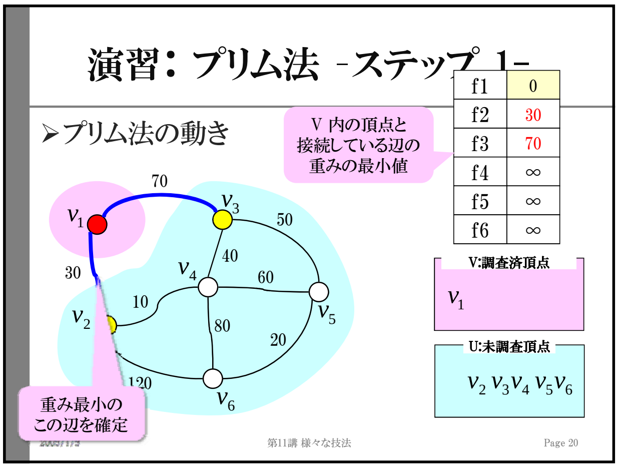

プリム法 - ステップ 3 -
この辺を確定
U から最小の distance を持 つ頂点を選ぶ
f1 0
f2 150
f3 200
f4 70
f5 160
f6 80
V:調査済頂点
$v_1 v_3\ \ v_6$
U:未調査頂点
$v_2\ \ v_4\ v_5$
2009/1/9
第11講 様々な技法
Page 15

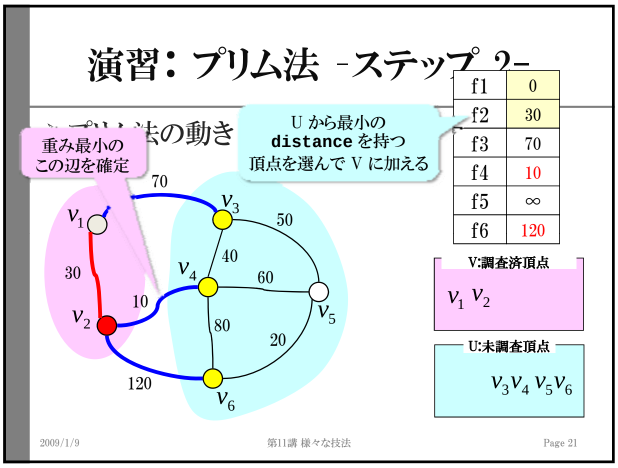

プリム法 - ステップ 4 -
この辺を確定
U から最小の distance を 持つ頂点を選ぶ
f1 0
f2 150
f3 200
f4 70
f5 160
f6 80
V:調査済頂点
$v_1 v_3 v_6\ \ v_4$
U:未調査頂点
$v_2\ \ v_5$
2009/1/9
第11講 様々な技法
Page 16

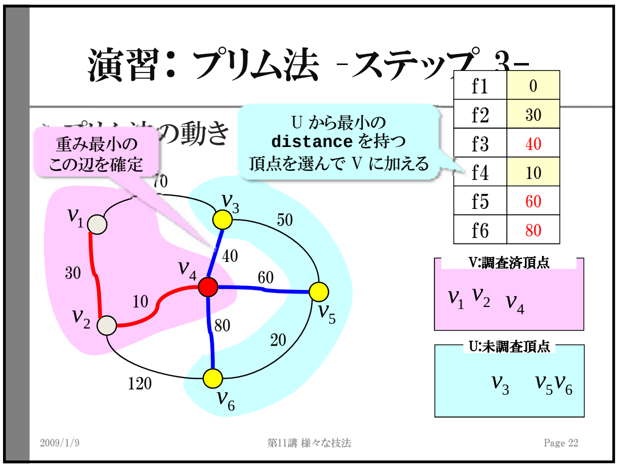

プリム法 - ステップ 5 -
この辺を確定
U から最小の distance を 持つ頂点を選ぶ
f1 0
f2 150
f3 200
f4 70
f5 160
f6 80
V:調査済頂点
$v_1 v_3 v_6\ \ v_4 v_2$
未調査頂点
v_5
2009/1/9
第11講 様々な技法
Page 17

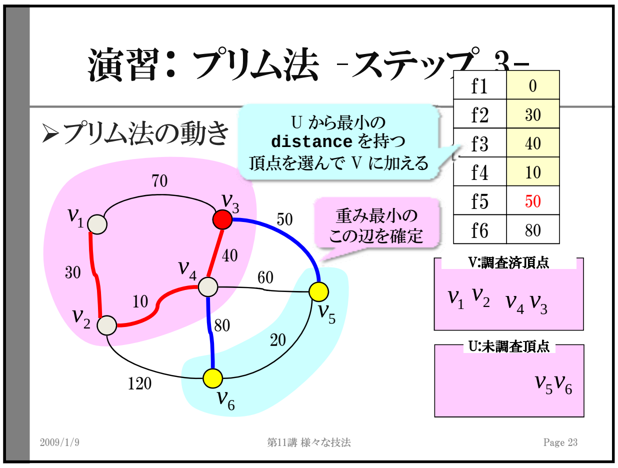

プリム法 - 終了 -
U から最小の distance を 持つ頂点を選ぶ
f1 0
f2 150
f3 200
f4 70
f5 160
f6 80
V:調査済頂点
$v_1 v_3 v_6 v_4 v_2 v_5$
未調査頂点
2009/1/9
第11講 様々な技法
Page 18

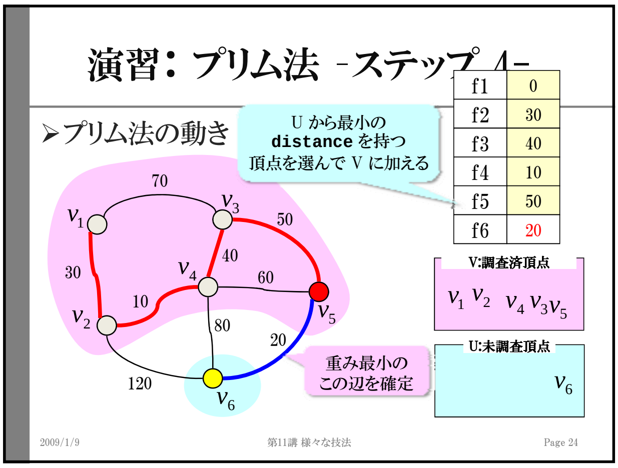

演習：プリム法
➢プリム法の動き
f1 0
f2 ∞
f3 ∞
f4 ∞
f5 ∞
f6 ∞
V:調査済頂点
U:未調査頂点
v1 v2 v3 v4 v5 v6
2009/1/9
第11講 様々な技法
Page 19


演習：プリム法 -ステップ 1-
➢プリム法の動き
V 内の頂点と
接続している辺の
重みの最小値
f1 0
f2 30
f3 70
f4 ∞
f5 ∞
f6 ∞
重み最小の
この辺を確定
V:調査済頂点
v1
U:未調査頂点
v2 v3 v4 v5 v6
第11講 様々な技法
Page 20


演習：プリム法 -ステップ 2-
重み最小の
この辺を確定
U から最小の
distance を持つ
頂点を選んで V に加える
f1 0
f2 30
f3 70
f4 10
f5 ∞
f6 120
V:調査済頂点
v1 v2
U:未調査頂点
v3 v4 v5 v6
2009/1/9
第11講 様々な技法
Page 21

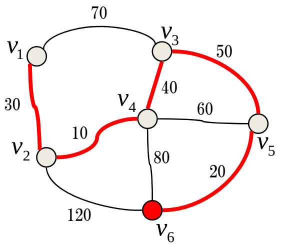

演習：プリム法 -ステップ 3-
重み最小の
この辺を確定
U から最小の
distance を持つ
頂点を選んで V に加える
f1 0
f2 30
f3 40
f4 10
f5 60
f6 80
V:調査済頂点
v1 v2 v4
U:未調査頂点
v3 v5 v6
2009/1/9
第11講 様々な技法
Page 22


演習：プリム法 -ステップ 3-
➢プリム法の動き
U から最小の
distance を持つ
頂点を選んで V に加える
重み最小の
この辺を確定
f1 0
f2 30
f3 40
f4 10
f5 50
f6 80
V:調査済頂点
v1 v2 v4 v3
U:未調査頂点
v5 v6
2009/1/9
第11講 様々な技法
Page 23


演習：プリム法 -ステップ 4-
➢プリム法の動き
U から最小の
distance を持つ
頂点を選んで V に加える
f1 0
f2 30
f3 40
f4 10
f5 50
f6 20
重み最小の
この辺を確定
V:調査済頂点
v1 v2 v4 v3 v5
U:未調査頂点
v6
2009/1/9
第11講 様々な技法
Page 24

演習：プリム法 -終了-

- プリム法の動き



f1	0
f2	30
f3	40
f4	10
f5	50
f6	20

V:調査済頂点

v_1 v_2 v_4 v_3 v_5 v_6

U:未調査頂点

本日の講義内容

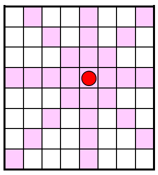
- バックトラック法
  - n 女王問題
- ゲーム木，ミニマックス法
- NP とは？
  - ナップサック問題，巡回セールスマン問題
- 分枝限定法，動的計画法，貪欲戦略，発見的手法，確率アルゴリズム

バックトラック法

- Backtracking，後戻り法
- しらみつぶしを効率的に行う方法
  - しらみつぶし
    - 全ての可能性を探索する方法
    - ゲームやパズルを計算機にさせるときに良く現れる
- 後戻りを繰り返しながら全ての可能性を辿るが，その時点で辿る必要のない（可能性のない）部分を辿ることは省略する
- バックトラック法はゲームやパズルの範囲に留まらない
  - 組合せ的な問題が少し複雑になるとバックトラック法が良く使われる
  - 組合せ的な問題
    - 解となる候補の数が，変数への値の組合せで爆発的に増える問題

n 女王問題（n-queen problem）

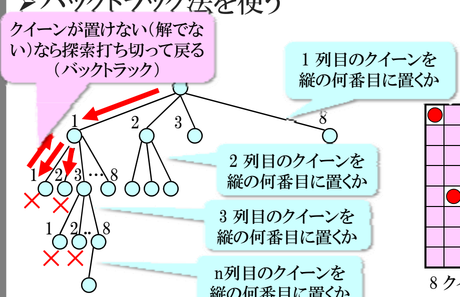
- $n \times n$ のチェスの盤面にクイーンをお互いの影響力のない部分に n 個置く問題
- n^2 個のマスからクイーンの位置を n 個選ぶので組合せは n^2 C n = $n^2!/(n! \times (n^2-n)!)$
  - これは多すぎるし，考えなさすぎている
- 同じ列には 2 つクイーンは置かないので，各列の縦 n マスのうちのどこに置くかを考えれば良い
  - これで $n!$ 通りに削減できる
- ではどうやって解くか？
  - しらみつぶし？

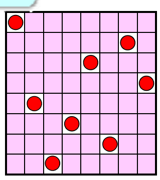


クイーンの影響範囲

n 女王問題の解法

- バックトラック法を使う

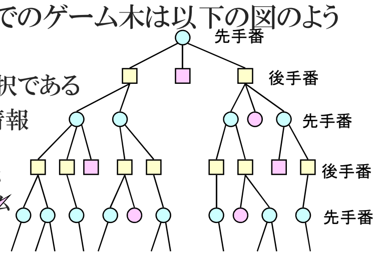


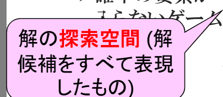


8 クイーンの解の一例

ゲーム木

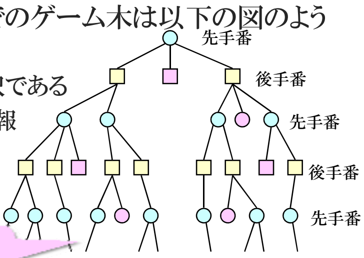
- 碁や将棋を計算機にさせることを考える
- 一般的な 2 人でのゲーム木は以下の図のように表せる
  - 枝分かれは選択である
  - ここでは完全情報ゲームを想定
    - 確率の要素が





ゲーム木

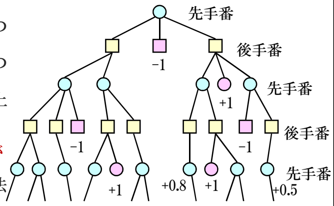
- 碁や将棋を計算機にさせることを考える
- 一般的な 2 人でのゲーム木は以下の図のように表せる
  - 枝分かれは選択である
  - ここでは完全情報
  - ゲームを想定
    - 確率の要素が
    - 入らないゲーム



解の探索空間
(解候補をすべて表現したもの)

ミニマックス法

- mini-max method
  - お互いが最善を尽くす仮定で探索を行う
  - 評価関数を設定
    - 先手の勝ち +1，引き分け 0，後手の勝ち −1
  - 先手は最大の子の値を持つ枝を辿る
  - 後手は最小の子の値を持つ枝を辿る
  - 手数が大きい場合は実用上ある深さで打ち切る
    - 途中局面での評価で判断
    - 局面の評価関数を作るのが難しい
  - プログラムはバックトラック法で動作



NP とは？

- やさしい問題とむずかしい問題
  - 効率の良いアルゴリズムがある(多項式時間で解ける)のがやさしい問題，
  - ないのがむずかしい問題
- 決定性アルゴリズムで多項式時間で解けるのがクラス P に属する問題
  - 決定性アルゴリズム：入力の値や内部の状態によって次の状態が一意に決まるアルゴリズム
- 非決定性アルゴリズムで多項式時間で解けるのがクラス NP に属する問題
  - 非決定性アルゴリズム：同じ入力でも次の状態が複数あるアルゴリズム，どちらを選んでも良いが，どちらが正しい選択かは分からない
  - 神が選択していけば多項式時間で答えが求まる

クラス NP に属する問題の例(1)

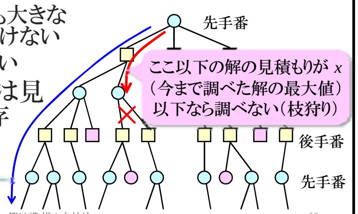
- ナップサック問題
  - 重さあたりの価値の異なる複数の物をできるだけ合計価値が高くなるようにナップサックに詰める問題
    - ナップサックには容量がある
    - 物は分割できない
    - 隙間ができてしまうなら，小さくて高価な物を詰めるより，単価が安くてもぴったり入る物を詰めた方がナップサック内の合計価値は上がるかもしれない
    - 神が選べば $O(n)$ で最適な選択ができる
      - ちなみに，物体が分割できるならクラス P になる
        - 単価の高いのから順に詰めるだけ
        - 最後の物はぴったりになるように削る

クラス NP に属する問題の例(2)

- 巡回セールスマン問題
  - TSP(Traveling Salesman Problem)
  - あるセールスマンが複数の都市を回って営業するとき，どの順に回れば最短経路で全ての都市を巡回できるか，最後に元の都市に戻ってくる
  - n 都市あるとして次の都市の候補は n-1
  - 全部で $n!$ 通りの組合せがある
  - とりあえず近い順に次の都市を選ぶと最初の都市に戻ってくるときに遠くなる
  - 神が選べば $O(n)$ で最適な巡回路が得られる

分枝限定法

- branch and bound
- 最適化問題をバックトラック法で解く手法
- 途中で楽観的な見積もりを出し，いままでの最大値よりも小さければ枝狩り
  - 絶対実際の値よりも大きな見積もりでないといけない
  - 悲観的ではいけない
- 削減できる計算量は見積もりの精度に依存
  - これがむずかしい



とりあえず 1 つ解を調べる(解の値 x)

動的計画法

- dynamic programming
- 部分的な解の状態を表の形で保存
- *k* 個の要素を取ったときの最適解が表で与えられているときに，*k*+1 個の要素を取ったときの最適解がどうなるかを表を元に計算，これを *n* まで続ける

例：長さ k の経路を求める問題の場合，長さ *k*+1 の経路を求める問題は，前の問題の結果を表にして残しておけば効率的に解ける

貪欲戦略

- greedy algorithm
- 現時点で最適な次の一手を選択
  - 一手先までしか先読みしない
- 複雑な問題の場合，局所的に最適な解が大局的にも最適とは保証ができない
- ダイクストラ法やプリム法がこれにあたる
  - この 2 つの方法はこれでうまくいく
- 多くの問題はあまり複雑ではないのでこの方法で良い解が得られることが多い

発見的方法

- heuristics
- 良い方法であるとは必ずしも証明されていないが，多くの場合でうまく行くことが知られているような手法
  - 理論よりも経験から得られた知識に頼った方法である
- 与えられたデータの傾向が既知であればそれを利用する例も多い
- 問題が難しくなると理論で片付けるのがむずかしくなるので良く用いられる
- GA（遺伝的アルゴリズム）がこれに当たる
  - 評価値の良い解を組合せて次の解を作ることを繰り返す

確率アルゴリズム

- probabilistic algorithm
- アルゴリズムの根本的な部分に確率が関係しているアルゴリズム
- ほとんどの場合は正しい解を返すが，ある一定の確率で間違えた解を出す計算法
  - アルゴリズムの定義「正当性」から外れるがこれもアルゴリズムと呼ばれる
- 計算量と精度はトレードオフ
  - 非常に早く解を出せることもあるが，精度は落ちる

第 11 講のまとめ

- バックトラック法
  - やり直しを伴いながら探索を進める方法
- ゲーム木，ミニマックス法
  - 探索空間の表現と，仮定による探索空間の限定
- NP とは？
  - むずかしい問題，神のみ簡単に解ける
- 分枝限定法，動的計画法，貪欲戦略，発見的手法，確率アルゴリズム
  - いろいろな方法があるなぁ

第 12 講の予告

- 模擬試験
  - 期末試験の模擬試験をします
- 質問会
  - 授業内容に関する質問疑問にお答えします
  - 今までの授業中，質問したかったけどできなかった人などは質問してください
- 試験について
  - 1月23日（金）4限
  - 持ち込みはミニレポートと自作ノートのみ（講義資料はダメ）
  - 今までのミニレポートの提出が半分以下の人は受験してもおそらく受かりません（満点などでない限り）