

アルゴリズムとデータ構造III 7回目:11月13日

グラフ

(動的計画法, ダイクストラ法, DPマッチング,
A*アルゴリズム)

授業資料 <http://ir.cs.yamanashi.ac.jp/~ysuzuki/algorithm3/index.html>

1

授業評価アンケート(中間期評価)

- 授業の最後に回収
- 授業科目番号:263216
- 授業科目名:アルゴリズムとデータ構造III

2

授業の予定(中間試験まで)

1	10/02	スタック(後置記法で書かれた式の計算)
2	10/09	チューリング機械, 文脈自由文法
3	10/16	構文解析 CYK法
4	10/23	構文解析 CYK法
5	10/30	構文解析(チャート法), グラフ(ダイクストラ法)
6	11/06	構文解析(チャート法), グラフ(ダイクストラ法, DPマッチング)
7	11/13	グラフ(DPマッチング, A*アルゴリズム)
8	11/20	グラフ(A*アルゴリズム), 前半のまとめ
9	11/27	中間試験

3

授業の予定(中間試験以降)

10	12/04	全文検索アルゴリズム(simple search, KMP)
11	12/11	全文検索アルゴリズム(BM, Aho-Corasick)
12	12/18	全文検索アルゴリズム(Aho-Corasick), データ圧縮
13	01/08	暗号(黄金虫, 踊る人形) 符号化(モールス信号, Zipfの法則, ハフマン符号)テキスト圧縮
14	01/15	テキスト圧縮(zip), 音声圧縮(ADPCM, MP3, CELP), 画像圧縮(JPEG)
15	01/29	期末試験

4

本日のメニュー

- 動的計画法
- ダイクストラ法(復習)
 - 動作例
 - アルゴリズム
- DPマッチング
 - 動作例
 - アルゴリズム
- A*アルゴリズム

5

動的計画法 (Dynamic Programming)

- 解くのに時間のかかる問題を、複数の部分問題に分割することで効率的に解くアルゴリズム

6

ダイクストラ法

- 動的計画法を最短経路問題に適用
- 最適経路中の部分経路もまた最適経路になっている

7

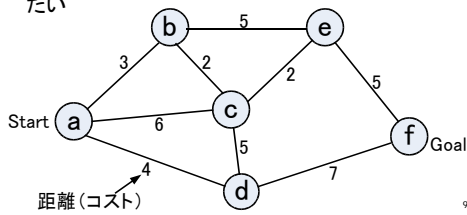
身近な最短経路問題

- 道路の経路探索(カーナビなど)

8

ダイクストラ法(最短経路問題用アルゴリズム)

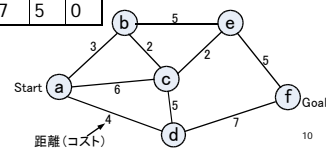
- StartノードからGoalノードへ最小コストで移動したい



9

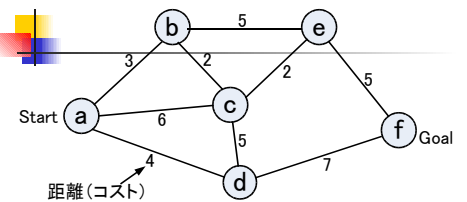
隣接行列(コスト付き)

	a	b	c	d	e	f
a	0	3	6	4	-	-
b	3	0	2	-	5	-
c	6	2	0	5	2	-
d	4	-	5	0	-	7
e	-	5	2	-	0	5
f	-	-	-	7	5	0



10

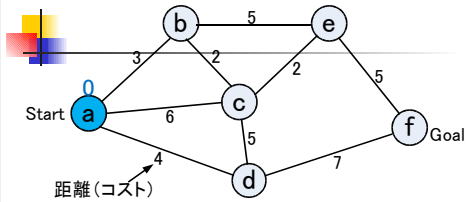
ダイクストラ法 動作例 1/13



- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定) — 確定済ノードからのアーク
- 7 Startからの最短距離(確定済) — 次期確定ノード決定に使用

11

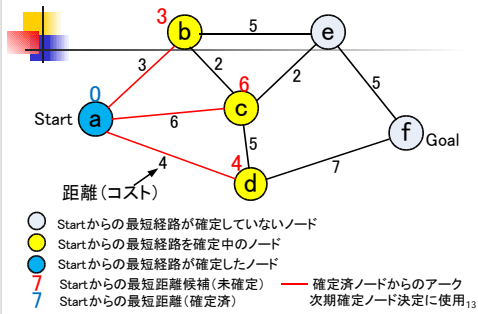
ダイクストラ法 動作例 2/13



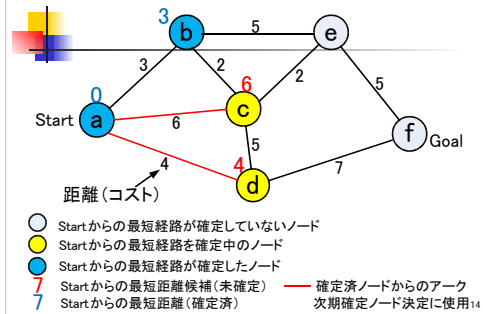
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定) — 確定済ノードからのアーク
- 7 Startからの最短距離(確定済) — 次期確定ノード決定に使用

12

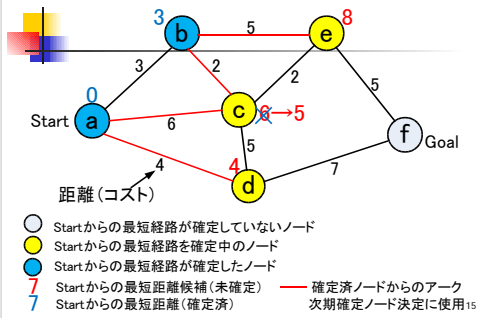
ダイクストラ法 動作例 3/13



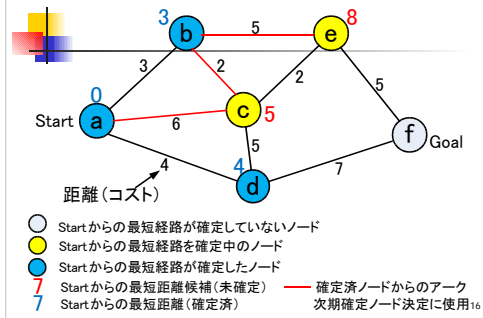
ダイクストラ法 動作例 4/13



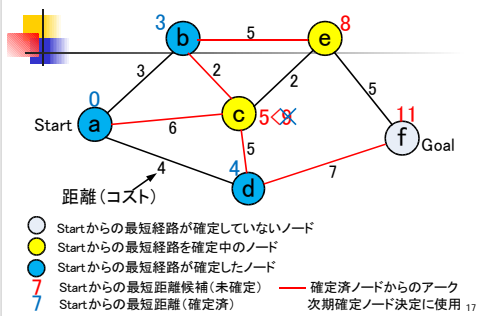
ダイクストラ法 動作例 5/13



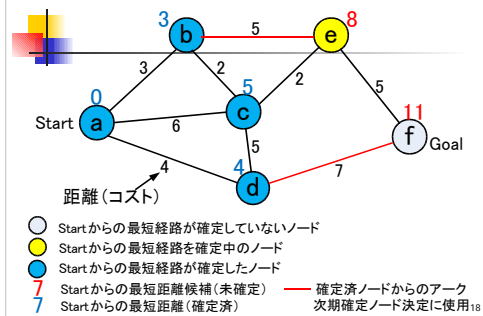
ダイクストラ法 動作例 6/13



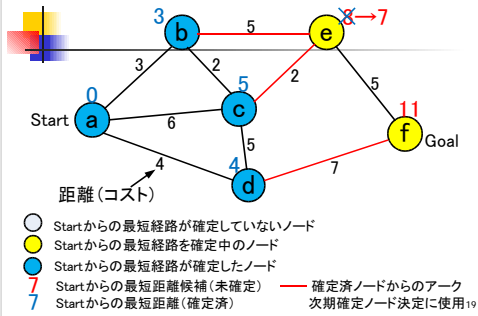
ダイクストラ法 動作例 7/13



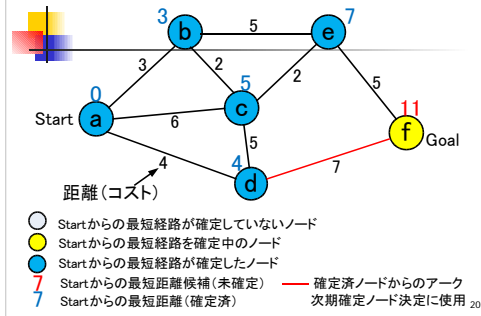
ダイクストラ法 動作例 8/13



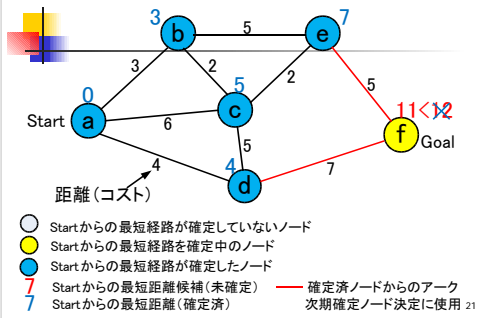
ダイクストラ法 動作例 9/13



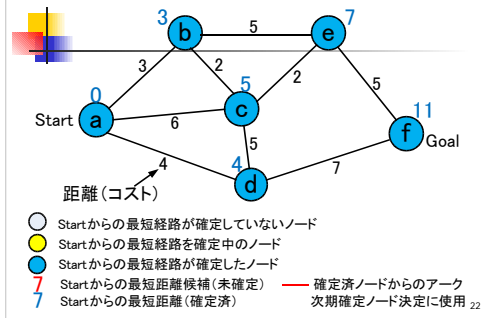
ダイクストラ法 動作例 10/13



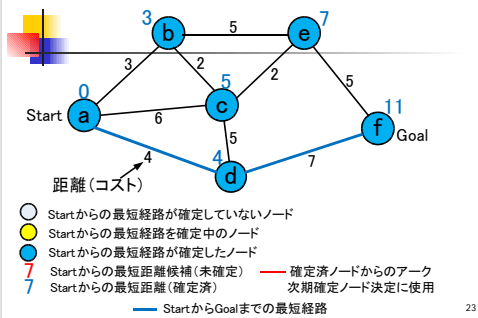
ダイクストラ法 動作例 11/13



ダイクストラ法 動作例 12/13



ダイクストラ法 動作例 13/13



ダイクストラ法 アルゴリズム

- 初期化: スタートノードの値 (最小コスト候補) を0, 他のノードの値を無限大に設定
- 未確定ノードが無くなるまで以下のループを繰り返す.
 - 確定中ノードのうち, 最小の値を持つノードを見つけ, 確定ノードとする.
 - 確定ノードからのエッジに対して「確定ノードまでのコスト + エッジのコスト」を計算し, そのノードの現在値よりも小さければ更新.

ダイクストラ法の特徴

- 最短経路の見つけ方
 - ゴールノードから「どこから来たのか」調べ、さかのぼる。
- マイナスのコストを持つエッジは扱えない。
- 特定のノードからの最短距離およびその経路が全てのノードに対して求まる。

25

DPマッチング

(例: 文字列の照合)

- 2つの文字列がどのくらい似ているかを調べる。
 - takeda は nakadai とどのくらい似ているか
 - 置換, 脱落, 挿入に対応
- 音声認識にも使える
 - 音声を文字列に変換した後, 登録単語と比較
 - (現在主流の)HMM(Hidden Markov Model)に拡張可能
- DNAの比較にも使える
 - A(アデニン), G(グアニン), C(シトシン), T(チミン)の並び方の比較
 - ACTGAGCATTとCTGGACTACGの比較

DPマッチング (例: 文字列の照合)

- 簡単に比較できる例
 - abcdef
 - abzdef
- Aに対して脱落, 挿入, 置換
 - A: abcdef
 - B: abdef
 - C: abccdef
 - D: abzdef

DPマッチング: 脱落, 挿入, 置換誤りを考慮して文字列照合可能

DPマッチング(例: 文字列の照合) 1/8

takeda と nakadai の照合

不一致コスト表

文字が一致 → 0
文字が不一致 → 3

	n	a	k	a	d	a	i
t	3	3	3	3	3	3	3
a	3	0	3	0	3	0	3
k	3	3	0	3	3	3	3
e	3	3	3	3	3	3	3
d	3	3	3	3	0	3	3
a	3	0	3	0	3	0	3

28

DPマッチング(例: 文字列の照合) 2/8

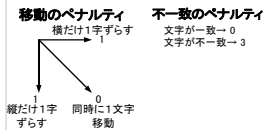
- takeda と nakadai の値を求める

不一致コスト表

	n	a	k	a	d	a	i
t	3	3	3	3	3	3	3
a	3	0	3	0	3	0	3
k	3	3	0	3	3	3	3
e	3	3	3	3	3	3	3
d	3	3	3	3	0	3	3
a	3	0	3	0	3	0	3

1文字ずらしたけれど文字が不一致: $1+3=4$ を加算

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7						
k	11						
e	15						
d	19						
a	23						



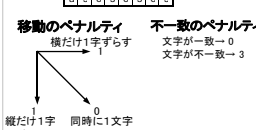
DPマッチング(例: 文字列の照合) 3/8

- takeda と nakadai の値を求める

不一致コスト表

	n	a	k	a	d	a	i
t	3	3	3	3	3	3	3
a	3	0	3	0	3	0	3
k	3	3	0	3	3	3	3
e	3	3	3	3	3	3	3
d	3	3	3	3	0	3	3
a	3	0	3	0	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11						
e	15						
d	19						
a	23						



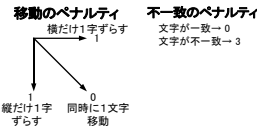
DPマッチング(例:文字列の照合) 4/8

takeda と nakadai の値を求める

不一致コスト表

n	a	k	a	d	a	i
t	3	3	3	3	3	3
a	3	0	3	3	3	3
k	3	0	3	3	3	3
e	3	3	3	3	3	3
d	3	3	3	0	3	3
a	3	3	3	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11	7	3	7	11	15	16
e	15						
d	19						
a	23						



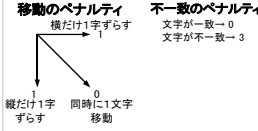
DPマッチング(例:文字列の照合) 5/8

takeda と nakadai の値を求める

不一致コスト表

r	a	k	a	d	a	i
t	3	3	3	3	3	3
a	3	0	3	3	3	3
k	3	0	3	3	3	3
e	3	3	3	3	3	3
d	3	3	3	0	3	3
a	3	3	3	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11	7	3	7	11	15	16
e	15	11	7	6	10	14	18
d	19						
a	23						



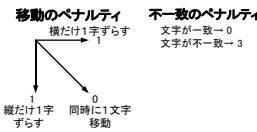
DPマッチング(例:文字列の照合) 6/8

takeda と nakadai の値を求める

不一致コスト表

n	a	k	a	d	a	i
t	3	3	3	3	3	3
a	3	0	3	3	3	3
k	3	0	3	3	3	3
e	3	3	3	3	3	3
d	3	3	3	0	3	3
a	3	3	3	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11	7	3	7	11	15	16
e	15	11	7	6	10	14	18
d	19	15	11	10	6	10	14
a	23						



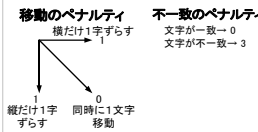
DPマッチング(例:文字列の照合) 7/8

takeda と nakadai の値を求める

不一致コスト表

r	a	k	a	d	a	i
t	3	3	3	3	3	3
a	3	0	3	3	3	3
k	3	0	3	3	3	3
e	3	3	3	3	3	3
d	3	3	3	0	3	3
a	3	3	3	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11	7	3	7	11	15	16
e	15	11	7	6	10	14	18
d	19	15	11	10	6	10	14
a	23	16	15	11	10	6	10



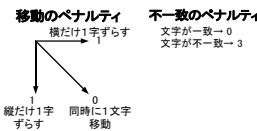
DPマッチング(例:文字列の照合) 8/8

takeda と nakadai の値を求める

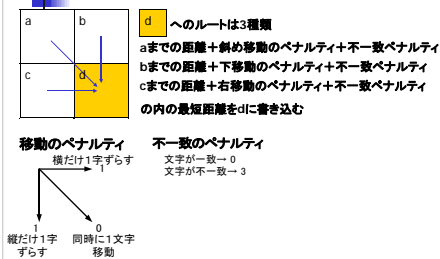
不一致コスト表

n	a	k	a	d	a	i
t	3	3	3	3	3	3
a	3	0	3	3	3	3
k	3	0	3	3	3	3
e	3	3	3	3	3	3
d	3	3	3	0	3	3
a	3	3	3	3	0	3

	n	a	k	a	d	a	i
t	3	7	11	15	19	23	27
a	7	3	7	8	12	13	17
k	11	7	3	7	11	15	16
e	15	11	7	6	10	14	18
d	19	15	11	10	6	10	14
a	23	16	15	11	10	6	10



アルゴリズム



DPマッチングの応用

- DPマッチングの探索空間を制限し、探索時間を削減する方法
 - ビームサーチ(最適解は保証されない)
 - A*アルゴリズム(最適解は保証される)
- HMM(隠れマルコフモデル)とビタビアルゴリズム
 - 音声認識手法の主流

37

A*アルゴリズム 最短経路探索問題

- ダイクストラ法にすこし工夫を加えた方法
- 各ノードからゴールまでの推定距離を利用
 - $0 \leq \text{推定距離} \leq \text{最短距離}$ でなければならない
 - 推定距離=0なら推定していないと同じ→ダイクストラ法

38