

## アルゴリズムとデータ構造III 8回目:11月25日

動的計画法, A\*アルゴリズム, DPマッチング

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

### 授業の予定(中間試験まで)

1	10/07	スタック(後置記法で書かれた式の計算)
2	10/14	チューリング機械, 文脈自由文法
3	10/21	構文解析 CYK法
4	11/04	構文解析 CYK法
5	11/11	構文解析(チャート法)
6	11/18	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	11/19 4時限 B2-41	グラフ(A*アルゴリズム, DPマッチング)
8	11/25	グラフ(DPマッチング), 前半のまとめ
9	12/02	中間試験

### 授業の予定(中間試験以降)

10	12/09	全文検索アルゴリズム(simple search, KMP)
11	12/16	全文検索アルゴリズム(BM, Aho-Corasick)
12	01/06	全文検索アルゴリズム(Aho-Corasick), データ圧縮
13	01/13	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	01/20	テキスト圧縮(zip), 音声圧縮(ADPCM, MP3, CELP), 画像圧縮(JPEG)
15	02/03	期末試験

### 中間試験

- 中間試験日
  - 12月02日(木)
- 範囲
  - スタック
  - 文脈自由文法
  - 構文解析
  - CYK法
  - (トップダウンチャート法)
  - 動的計画法
  - ダイクストラ法
  - A\*アルゴリズム
  - DPマッチング

### 本日のメニュー

- 動的計画法
- A\*アルゴリズム(復習)
- DPマッチング
  - アルゴリズム
  - 動作
- 中間試験の範囲の説明

### 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

## 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

## 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(距離更新時に直前のノードを記述しておく)
  - 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
- 情報処理技術者試験で頻出
  - 平成15年秋期 基本情報技術者試験 午後の問題
  - 応用情報技術者試験にも出題

## 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

## ダイクストラ法のアルゴリズム

- ```

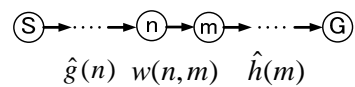
begin
  for each x ∈ V do begin
    cost[x] := w[s, x];
    parent[x] := 's';
  end
  U := V - {s};
  while U ≠ ∅ do
    begin
      U中のmで, cost[m]が最小となる頂点mを選ぶ;
      U := U - {m};
      mから隣接する頂点の集合をDmとする;
      for each x ∈ Dm ∩ U do
        If cost[m] + w[m, x] < cost[x]
          then begin
            Cost[x] := cost[m] + w[m, x];
            Parent[x] := m;
          end
      end
    end
  end
    
```
- costとparentの初期化
- U(未確定ノード)の初期化
- 集積コストが最も小さいノードmを選んで, cost[m]とparent[m]を確定
- 頂点mから隣接するノードすべての集合D<sub>m</sub>を求める。D<sub>m</sub>の要素で且つ未確定ノードである各xについてmを経由してxに至る最短経路のコストを計算し, 現在のcost[x]と比較し, 小さければ更新する

## 最短経路問題(より効率の良いアルゴリズム)

- ダイクストラ法は各ノードとスタートの間のコストだけに注目
- もし各ノードとゴールの間のコストが予想できれば効率よく探索可能

## A\*アルゴリズム

- 評価関数  $\hat{f}(m)$  が小さい順に経路を探索



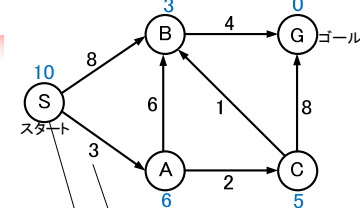
$$\hat{f}(m) = \hat{g}(n) + w(n, m) + \hat{h}(m)$$

$$\text{ただし } 0 \leq \hat{h}(m) \leq h(m)$$

## 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

## まずはAアルゴリズム



Aを経由するルートでの推定コスト評価値  

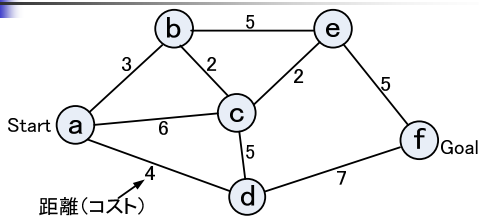
$$\hat{f}(A) = \hat{g}(S) + w(S, A) + \hat{h}(A)$$

$$= 0 + 3 + 6 = 9$$

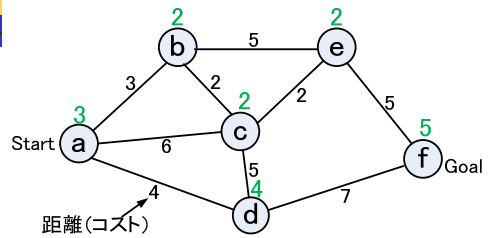
AからGまでの推定コスト

$\forall n, 0 \leq \hat{h}(n) \leq h(n)$   
 を満たすとき  
 A\*アルゴリズム  
 但し、 $n$ は節点番号、  
 $h(n)$ は節点 $n$ から  
 ゴールまでのコスト

## A\*アルゴリズム 動作例

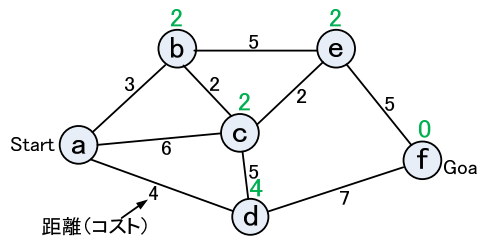


## A\*アルゴリズム 動作例 1/14



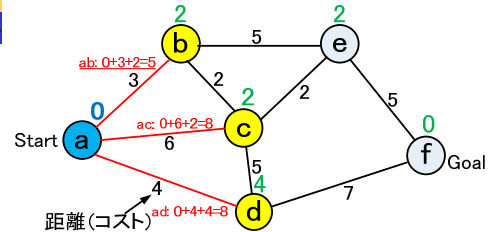
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

## A\*アルゴリズム 動作例 2/14



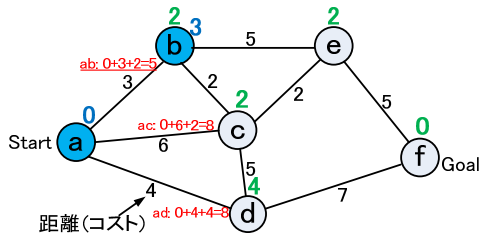
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

## A\*アルゴリズム 動作例 3/14



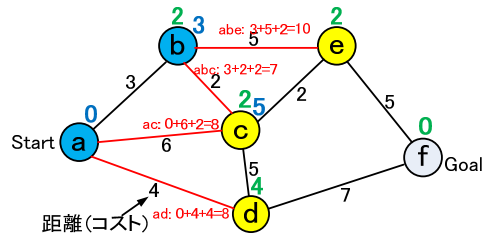
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム 動作例 4/14



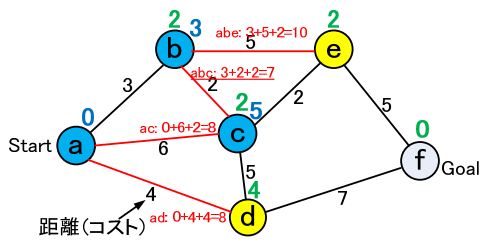
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 5/14



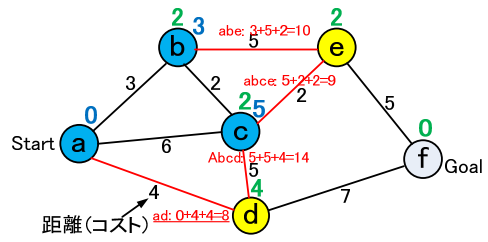
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 6/14



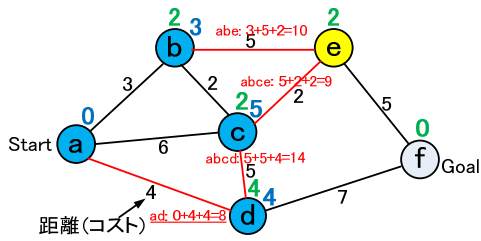
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 7/14



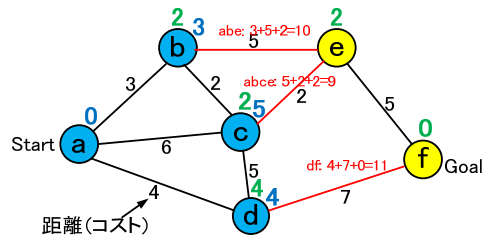
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 8/14



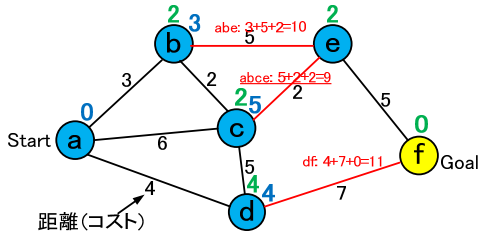
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 9/14



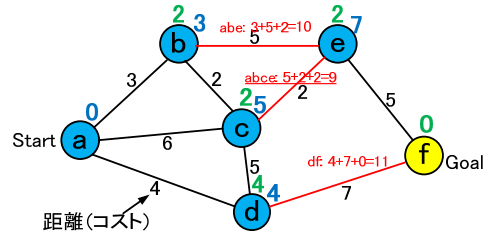
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 10/14



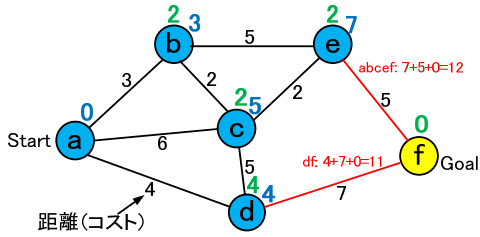
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 11/14



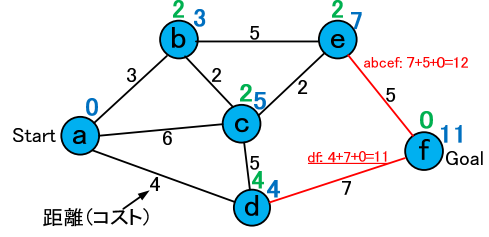
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 12/14



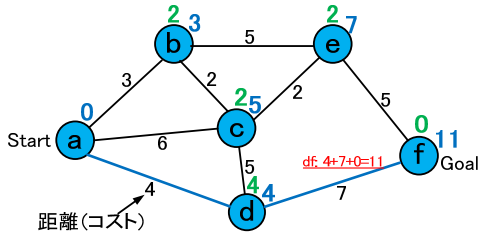
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 13/14



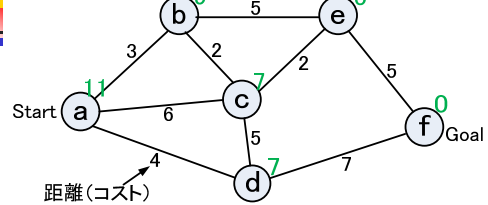
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム 動作例 14/14



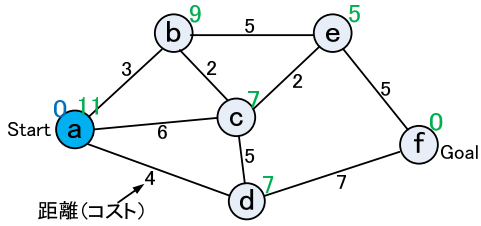
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム (最良の $\hat{h}(m)$ ) 1/6



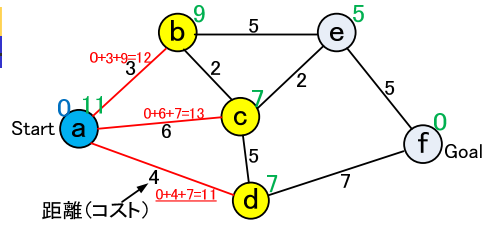
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

### A\*アルゴリズム その2 2/6



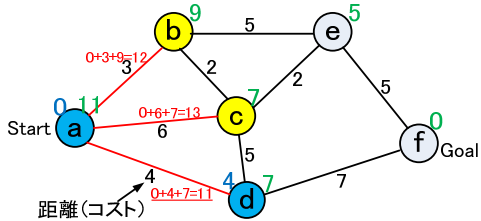
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム その2 3/6



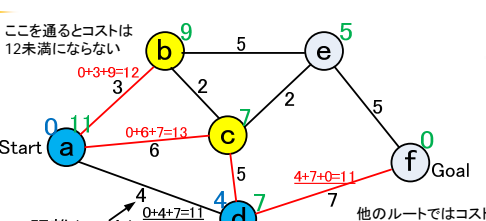
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム その2 4/6



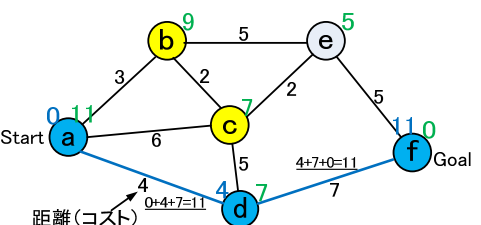
- Startからの最短経路が確定していないノード
- Startからの最短経路を確定中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム その2 5/6



- 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)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム その2 6/6



- 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)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 次期確定ノード決定に使用
- StartからGoalまでの最短経路

### A\*アルゴリズム, A\*アルゴリズム ダイクストラ法

- 評価式  $f(n) = g(n) + h(n)$ 
  - n: 節点番号
  - $f(n)$ : 節点nを通りスタートからゴールまでの距離
  - $g(n)$ : スタートから節点nまでの距離
  - $h(n)$ : 節点nからゴールまでの距離
- Aアルゴリズム
  - $f^*(n) = g(n) + h^*(n)$ を利用してスタートからゴールまでの距離を調べる
    - $f^*(n)$ : 節点nからゴールまでの距離の評価値
    - $h^*(n)$ : 節点nからゴールまでの距離の評価値
- A\*アルゴリズム
  - Aアルゴリズムの $f^*(n) = g(n) + h^*(n)$ の $h^*(n)$ が $0 \leq h^*(n) \leq h(n)$ の時
  - 最初に見つけたルートが最短ルートであることが保証されている
- ダイクストラ法
  - Aアルゴリズムの $f^*(n) = g(n) + h^*(n)$ の $h^*(n)$ が0の時
  - つまり $f^*(n) = g(n)$

## 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

前回はここまで

## 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: abcdef に対して脱落, 挿入, 置換
  - A: abcdef
  - B: abdef (脱落)
  - C: abccdef (挿入)
  - D: abzdef (置換)

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

文字が一致 → 0  
文字が不一致 → 3

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 |

## 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

移動のペナルティ  
横だけ1文字ずらす → 1  
縦だけ1文字ずらす → 1  
同時に1文字移動 → 0

不一致のペナルティ  
文字が一致 → 0  
文字が不一致 → 3

|   |    |   |    |    |    |    |    |
|---|----|---|----|----|----|----|----|
|   | 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 |

移動のペナルティ  
横だけ1文字ずらす → 1  
縦だけ1文字ずらす → 1  
同時に1文字移動 → 0

不一致のペナルティ  
文字が一致 → 0  
文字が不一致 → 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 | 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 |

移動のペナルティ  
横だけ1文字ずらす → 1  
縦だけ1文字ずらす → 1  
同時に1文字移動 → 0

不一致のペナルティ  
文字が一致 → 0  
文字が不一致 → 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 |   |    |    |    |    |    |

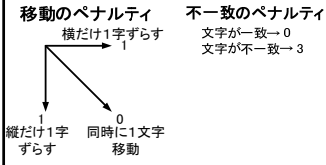
### DPマッチング(例:文字列の照合) 5/8

- takeda と nakadai の値を求める

不一致コスト表

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | 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 | 7  | 3  | 7  | 11 | 15 | 16 |
| e | 15 | 11 | 7  | 6  | 10 | 14 | 18 |
| d | 19 |    |    |    |    |    |    |
| a | 23 |    |    |    |    |    |    |



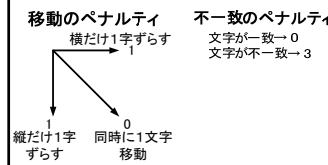
### DPマッチング(例:文字列の照合) 6/8

- 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 | 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 | 0 | 3 | 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 | 7  | 3  | 7  | 11 | 15 | 16 |
| e | 15 | 11 | 7  | 6  | 10 | 14 | 18 |
| d | 19 | 15 | 11 | 10 | 6  | 10 | 14 |
| a | 23 |    |    |    |    |    |    |



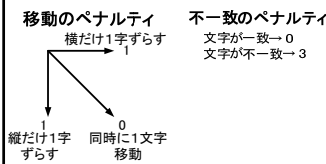
### DPマッチング(例:文字列の照合) 7/8

- takeda と nakadai の値を求める

不一致コスト表

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | 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 | 0 | 3 | 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 | 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 |



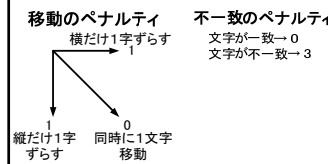
### DPマッチング(例:文字列の照合) 8/8

- takeda と nakadai の値を求める

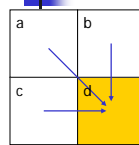
不一致コスト表

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | n | a | k | a | d | a | i |
| t | 3 | 3 | 3 | 3 | 3 | 3 | 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 | 0 | 3 | 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 | 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) アルゴリズム



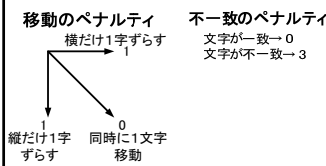
d へのルートは3種類

aまでの距離+斜め移動のペナルティ+不一致ペナルティ

bまでの距離+下移動のペナルティ+不一致ペナルティ

cまでの距離+右移動のペナルティ+不一致ペナルティ

内の最短距離をdに書き込む



### DPマッチング 練習問題 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し、不一致ペナルティは3点、挿入ペナルティ=1、脱落ペナルティ=1とする。

通行ペナルティ積算表

|   |    |    |   |    |
|---|----|----|---|----|
|   | a  | b  | c | d  |
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | ①  |
| c | 8  | 7  | ② | ③  |
| d | 12 | 11 | ④ | ⑤  |



### DPマッチング 練習問題 解答1/5 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し, 不一致ペナルティは3点, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

|   | a  | b  | c | d  |
|---|----|----|---|----|
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | 8  |
| c | 8  | 7  | ② | ③  |
| d | 12 | 11 | ④ | ⑤  |

### DPマッチング 練習問題 解答2/5 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し, 不一致ペナルティは3点, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

|   | a  | b  | c | d  |
|---|----|----|---|----|
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | 8  |
| c | 8  | 7  | ③ | ③  |
| d | 12 | 11 | ④ | ⑤  |

### DPマッチング 練習問題 解答3/5 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し, 不一致ペナルティは3点, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

|   | a  | b  | c | d  |
|---|----|----|---|----|
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | 8  |
| c | 8  | 7  | 3 | 7  |
| d | 12 | 11 | ④ | ⑤  |

### DPマッチング 練習問題 解答4/5 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し, 不一致ペナルティは3点, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

|   | a  | b  | c | d  |
|---|----|----|---|----|
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | 8  |
| c | 8  | 7  | 3 | 7  |
| d | 12 | 11 | ④ | ⑤  |

### DPマッチング 練習問題 解答5/5 (昨年度中間試験より)

- 下の表は「abcd」と「accd」の単語間距離をDPマッチングにより計算しているところである。
- 表中の①, ②, ③, ④, ⑤には何が入るか答えよ。
- 但し, 不一致ペナルティは3点, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

|   | a  | b  | c | d  |
|---|----|----|---|----|
| a | 0  | 4  | 8 | 12 |
| c | 4  | 3  | 4 | 8  |
| c | 8  | 7  | 3 | 7  |
| d | 12 | 11 | 7 | ③  |

### 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
  - A\*アルゴリズム
- 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

## 12月2日は中間試験

- 試験範囲は今日の授業まで
  - スタック
  - 文脈自由文法
  - 構文解析
  - CYK法
  - (トップダウンチャート法)
  - ダイクストラ法
  - A\*アルゴリズム
  - DPマッチング
- 試験問題の傾向はWebに掲載している去年の試験問題を見てください