データ構造とアルゴリズムII

第10回
単一始点最短路 (II)／全点対最短路

472

トポロジカル･ソート順による緩和

473

トポロジカル･ソート順に緩和

- 閉路のない有向グラフ限定
- 閉路がないならトポロジカル･ソート順に緩和するのがベルマン･フォードより速い
  - $\Theta(V + E)$

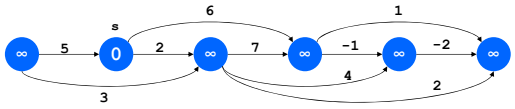
474

方針

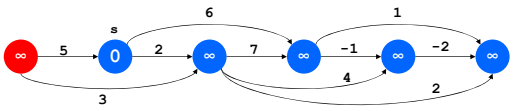
- グラフをトポロジカル･ソートして頂点に線形順序を与える
- ソート順に頂点を選び，その頂点の出辺を緩和する
  - 各頂点は一回だけ選択される

475

アルゴリズムの動き



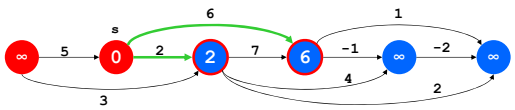
閉路のない有向グラフをトポロジカル・ソートする

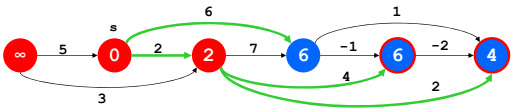


左の頂点から順番に選択，選択された頂点の出辺を緩和する

476

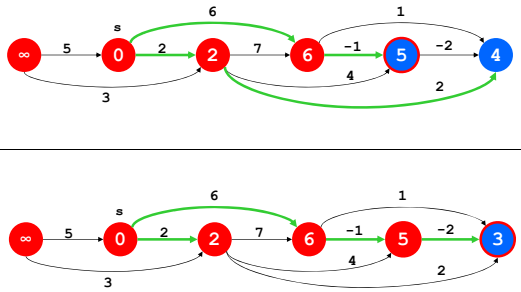
アルゴリズムの動き





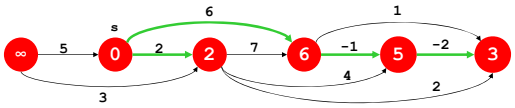
477

アルゴリズムの動き



478

アルゴリズムの動き



479

擬似コード

DAG-SHORTEST-PATHS(G, w, s)
  G の頂点をトポロジカル·ソートする
  INITIALIZE-SINGLE-SOURCE(G, s)
  for 各頂点 u (トポロジカル・ソート順に)
    do for 各頂点 v ∈ Adj[u]
      do RELAX(u, v, w)

480

アルゴリズムの正当性

- なぜこれで最短路木が得られるのか?
  - 証明すべきこと … "アルゴリズム終了時点で, すべての頂点 v に対して d[v] = δ(s, v) になっている"
    - これが分かれば先行点部分グラフの性質から最短路木が得られることが証明できる
  - 経路緩和性で証明 (補題 24.5)
    - 頂点をトポロジカルソート順に処理して緩和するから, 経路緩和性が成立する
      - 詳細は教科書参照のこと

481

計算量

- Θ(V + E)
  - トポロジカル・ソート Θ (V + E)
  - 初期化 O(V)
  - for ループの繰り返し回数 |E|, ループ一回あたり Θ(1) ・・・Θ(E)

482

ダイクストラのアルゴリズム

483

ダイクストラのアルゴリズム

- 全ての辺の重みが非負であるグラフの単一始点最短路問題を高速に解く
  - $w(u, v) \geqq 0$
- ベルマン･フォードのアルゴリズムより高速に解ける

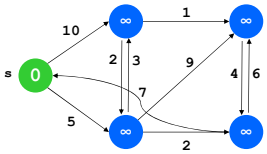
484

ダイクストラ法の方針

- 最小の最短路推定値を持つ頂点 u を選択し，u の出辺を緩和･･･を繰り返す
  - 貪欲戦略
  - 一度選択した頂点は二度と選択されない
    - 始点 s からの最終的な最短路重みが既に決まっている頂点の集合 S を管理
      - $u \in V - S$
    - u を選択 → u を S に追加 → u の出辺を緩和
    - u の選択に min 優先度付きキュー Q を用いる

485

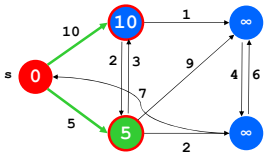
ダイクストラのアルゴリズムの動き



Q = V - S にある

選択された u (最小の最短路推定値)

S にある

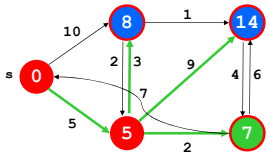


最初に選択された始点 s からの出辺が緩和される

始点 s は S に加えられる．次に選択されるのは，S にまだ入ってないもので最小の最短路推定値 (5) を持つ頂点

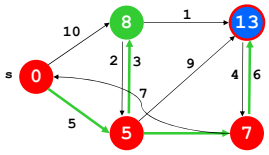
486

ダイクストラのアルゴリズムの動き



同様に u からの出辺が緩和される

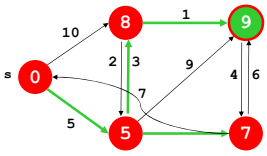
そして Q から次の u を選ぶ



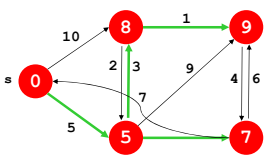
同様に繰り返す

487

ダイクストラのアルゴリズムの動き



緩和



最短路重み d, 最短路木 π がそれぞれ得られる

488

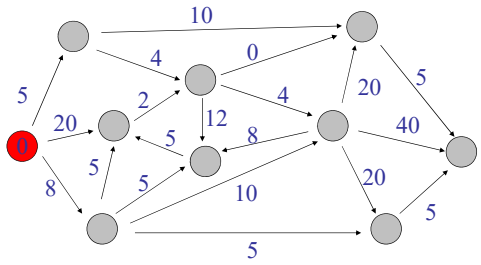
擬似コード

```
DIJKSTRA(G, w, s)
    INITIALIZE-SINGLE-SOURCE(G, s)
    S ← φ
    Q ← V[G]
    while Q ≠ φ
        do u ← EXTRACT-MIN(Q)
            S ← S ∪ {u}
            for 各頂点 v ∈ Adj[u]
                do RELAX(u, v, w)
```

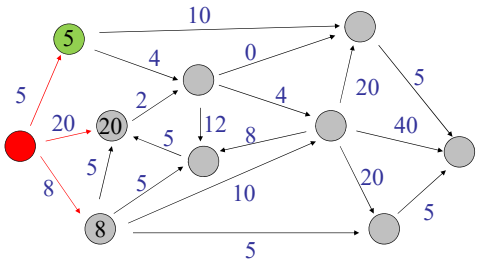
489

演習

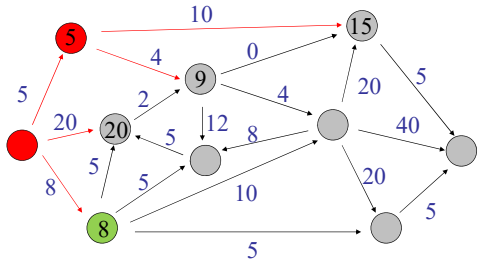
- このネットワークの赤の頂点からの最短距離をダイクストラのアルゴリズムで求めよ



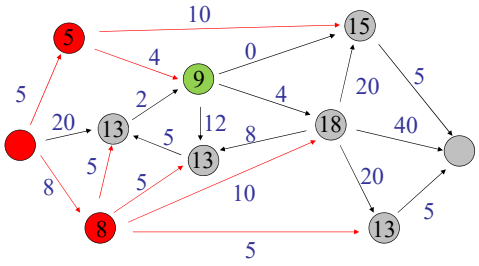
演習



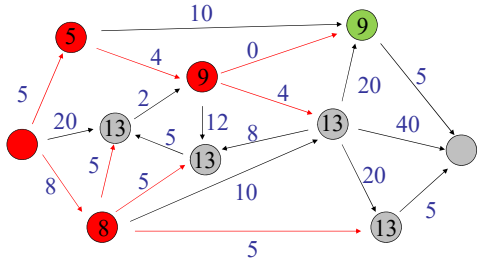
演習



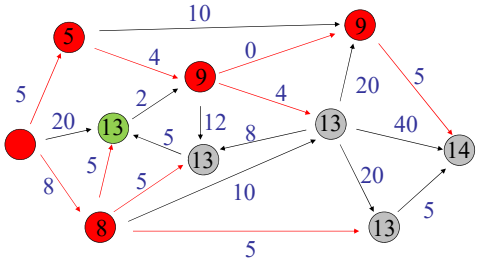
演習



演習



演習



アルゴリズムの正当性

- なぜこれで最短路木が得られるのか?
  - 証明すべきこと ･･･ "停止した後，すべての頂点 v に対して d[v] = δ(s, v) になっている"
    - これが分かれば先行点部分グラフの性質から最短路木が得られることが証明できる
  - while ループの各繰り返しの開始直前に既に d[v] = δ(s, v) が各頂点 v ∈ S について成立している (定理 24.6) ことから証明できる
    - 貪欲戦略に基づく選択が正しい

496

計算量

- min 優先度付きキューの実現方法に依存
  - EXTRACT-MIN が |V| 回
  - DECREASE-KEY が |E| 回
- 二分木ヒープ
  - EXTRACT-MIN, DECREASE-KEY 共に O(lg V)
  - → ダイクストラ法 O((V+E) lg V)
- フィボナッチヒープ
  - EXTRACT-MIN ･･･ O(lg V) ※ならし
  - DECREASE-KEY ･･･ O(1) ※ならし
  - → ダイクストラ法 O(VlogV + E)

497

まとめ

- 単一始点最短路問題を解くアルゴリズムについて学んだ
  - ベルマン･フォードのアルゴリズム
  - 閉路なし有向グラフ (トポロジカル･ソート)
  - ダイクストラのアルゴリズム
- 各頂点の選択順序，緩和操作の回数がそれぞれ異なる．またそれにより汎用性/効率性が変わる

498

差分制約と最短路

499

差分制約式系

- 制約が二つの変数の差に関して与えられる
- Ax ≦ b という行列の表現で書くと，Aの各行はちょうど一つの1と一つの-1を含み，残りは0

```
x1 - x2 ≦ 5
x1 - x3 ≦ 6
x2 - x4 ≦ -1
x3 - x4 ≦ -2
x4 - x1 ≦ -3
```

500

差分制約式系の適用分野

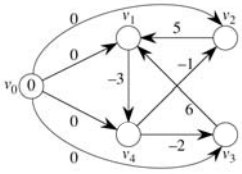
1. 各変数は，ある事象が生起する時刻を示す
   - 事象が生起する時刻の間隔に対して，ある値以下という制約が与えられる
2. 各変数は，電子回路上の端子の電位を表す
   - 端子間の電位差に関して，ある値以下という制約が与えられる

501

差分制約式系と最短経路 (1)

- 差分制約式系中の各変数x_1, …, x_nに対応する頂点v_1, …, v_nを考える
- さらに，始点となる特別な頂点v_0を導入する
- 始点と各頂点を結ぶ有向辺を導入，重みは0
- 制約 $x_j - x_i \leqq b$ に対して，辺 (v_i, v_j) を導入，重みはb

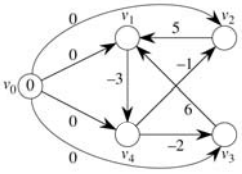
$$x1 - x2 \leqq 5$$
$$x1 - x3 \leqq 6$$
$$x2 - x4 \leqq -1$$
$$x3 - x4 \leqq -2$$
$$x4 - x1 \leqq -3$$



差分制約式系と最短経路 (2)

- この制約グラフの単一始点最短路を求めたとき，その最短路重みの値のベクトルが差分制約式系の解 (のひとつ) になる
- ベルマン・フォードのアルゴリズムが使える⇒ O(VE) すなわち多項式時間で解ける
- 演習：右のグラフの最短経路重みをベルマン・フォードのアルゴリズムを用いて求めよ

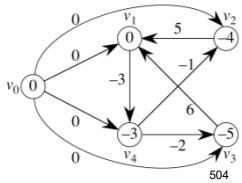
$$x1 - x2 \leqq 5$$
$$x1 - x3 \leqq 6$$
$$x2 - x4 \leqq -1$$
$$x3 - x4 \leqq -2$$
$$x4 - x1 \leqq -3$$



差分制約式系と最短経路 (3)

- 解答：(0, -4, -3, -5).
- ベクトルの各要素に同じ値を加えても制約を満たす．
  例えば (5, 1, 0, 2)
- 負の重みの閉路があると解が存在しない

$$x1 - x2 \leqq 5$$
$$x1 - x3 \leqq 6$$
$$x2 - x4 \leqq -1$$
$$x3 - x4 \leqq -2$$
$$x4 - x1 \leqq -3$$



504

25. 全点対最短路

505

25章の内容

- 動的計画法＋反復自乗法
- Floyd-Warshallのアルゴリズム
- Johnsonのアルゴリズム

506

最短経路の性質

$(v_i, \ldots, v_k, v_j)$がv_iからv_jへの最短経路であるとする．すなわち，最短路はv_jに到達する直前にv_kを経由する．この場合，以下が成立する．
ただし$\delta(v_i, v_j)$はv_i, v_j間の最短経路の長さ：

- $\delta(v_i, v_j)=\delta(v_i, v_k)+w(v_k, v_j)$

507

最短経路長を求める方法

- 動的計画法を使う:
  $d^{(m)}(i, j)$を，長さ(パス中の頂点の数)が高々mのi→j間の最短経路とする．以下のように帰納的に定義可能．

$$d^{(0)}(i,j)=\begin{cases}0 & ,i=j\\ \infty & ,i\neq j\end{cases}$$

$$d^{(m)}(i,j)=\min_{1\le k\le n}\{d^{(m-1)}(i,j),d^{(m-1)}(i,k)+w(k,j)\}$$

508

最短経路長の求め方

- n=|V|とすると，最短経路は閉路を含まないので，$d^{(n-1)}(i, j)$がi → j間の最短経路長となる．
- O(|V|³log|V|)で実行できる．
  Floyd-Warshallアルゴリズムによって，O(|V|³)に改善可能．

509

全点対最短路アルゴリズム

- 入力:　負の重みがない有向グラフ G=(V, E)，|V|=n.
  n×n の隣接行列 W=(W[i, j])

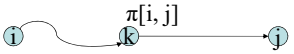
$$W[i,j]=\begin{cases}0 & ,i=j\\ w(i,j) & ,(u,v)\in E\\ \infty & ,(u,v)\notin E\end{cases}$$

510

全点対最短路アルゴリズム

出力:

- n×n の距離行列 D=(D[i, j])
  D[i, j]=δ(i, j)
- n×n の先行点行列 π=(π[i, j])
  もしi → jの経路がなければπ[i, j]=NIL，そうでなければ，k=π[i, j]の時，i → jの最短経路はkから直接jに至る．



511

```
Extend-Shortest-Paths(D,W)
{ Let D' = (D'[i,j]) be an n×n matrix
  for i=1 to n do
    for j=1 to n do D'[i,j]←∞
      for k=1 to n do
        D'[i,j]←min(D'[i,j],D[i,k]+W[k,j])
  return D'
}
```

Time Complexity: $O(n^3)$

512

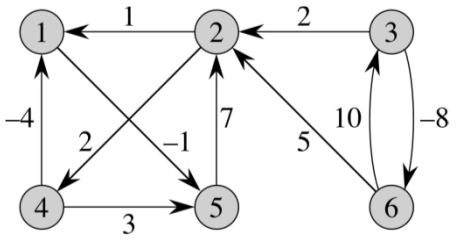
```
Slow-All-Pairs-Shortest-Paths(G,W)
{ D(1)←W
  for m=2 to n-1 do
   D(m)←Extend-Shortest-Paths(D(m-1),W)
  return D(n-1)
}
```

Time Complexity: $O(n^4)$

513

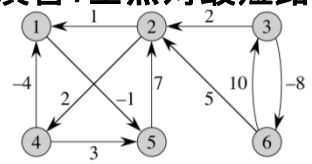
演習：全点対最短路

- 動的計画法で以下のグラフの全点対最短路を求めよ



514

演習：全点対最短路



$$W = D^{(1)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$$

515

演習：全点対最短路

$$W = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} \quad D^{(1)} = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 6 & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & 0 & \infty \\ 3 & -3 & 0 & 4 & \infty & -8 \\ -4 & 10 & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & \infty & 0 \end{pmatrix}$$

516

演習：全点対最短路

$$W = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} \quad D^{(2)} = \begin{pmatrix} 0 & 6 & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & 0 & \infty \\ 3 & -3 & 0 & 4 & \infty & -8 \\ -4 & 10 & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & \infty & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -2 & -3 & 0 & -1 & 2 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$$

517

演習：全点対最短路

$$W = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} \quad D^{(3)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -2 & -3 & 0 & -1 & 2 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -3 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

518

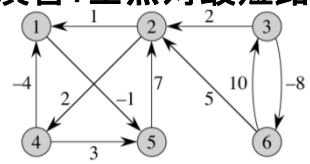
演習：全点対最短路

$$W = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} \quad D^{(4)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -3 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

519

演習：全点対最短路



$$D^{(5)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

520

反復二乗法

- 行列の積を求めるなら，行列Dのm乗 ($m = 2m'$) を求めるのに，Dをm回かけても，Dのm'乗を求めて，これを自乗しても結果は同じ．
- 同様に，Extend-Shortestot-Pathsの引数を，D^{m-1}とWではなく，両方を$D^{m'}$とすると，正しくD^mが得られる（経由する頂点数がm'以下の経路を二つ組み合わせれば，頂点数がm以下の経路が得られる）
- 最終的に，$n-1 \leq m'$になればOK.

521

```
Faster-All-Pairs-Shortest-Paths(G,W)
{ D(1)=W
  m=1
  while n-1>m do
    D(2m)←Extend-Shortest-Paths(D(m),D(m))
    m = 2m
  return D(m)
}
```

Time Complexity: $O(n^3 \log n)$

522

Floyd-Warshallアルゴリズム

- $O(n^3)$のアルゴリズム，負辺は存在しても良いが，負閉路はないと仮定．
- 最短路の中間頂点を考える．i➔j間の経路が$(i, u_1, \ldots, u_m, j)$とすると，$u_1, \ldots, u_m$が中間頂点．

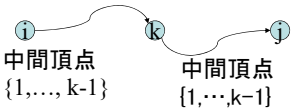
523

Floyd-Warshallアルゴリズム

- 頂点集合V={1, …, n}に関して，$d^{(k)}(i, j)$を，中間頂点として{1, …, k}のみが利用可能な場合のi➔j間の最短経路長とする．

$$d^{(0)}(i,j) = \begin{cases} 0 & ,i = j \\ w(i,j) & ,i \neq j \end{cases}$$

$$d^{(k)}(i,j) = \min\{d^{(k-1)}(i,j), d^{(k-1)}(i,k) + d^{(k-1)}(k,j)\}$$



524

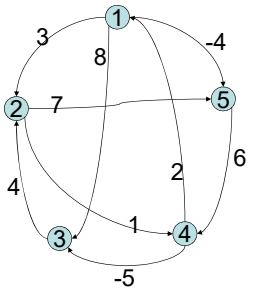
Floyd-Warshallアルゴリズム

```
Floyd-Warshall(G,W)
{ D(0)←W
  for k = 1 to n do
    for i = 1 to n do
      for j = 1 to n do
        if D(k-1)[i,j]>D(k-1)[i,k]+D(k-1)[k,j]
        then D(k)[i,j]←D(k-1)[i,k]+D(k-1)[k,j]
             π[i,j]← π[k,j]
        else D(k)[i,j]←D(k-1)[i,j]
  return D(n)
}
```

Time Complexity: $O(n^3)$

525

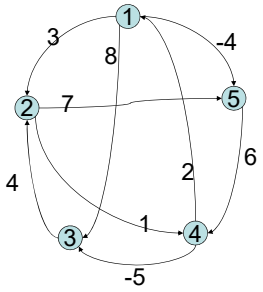
Floyd-Warshallの実行例



526

Floyd-Warshallの実行例

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

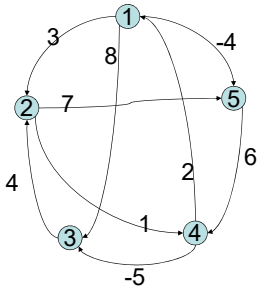


527

Floyd-Warshallの実行例

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

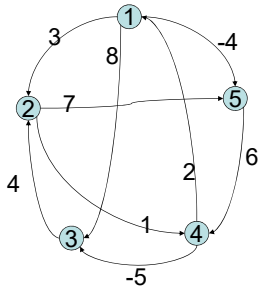
$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$



Floyd-Warshallの実行例

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

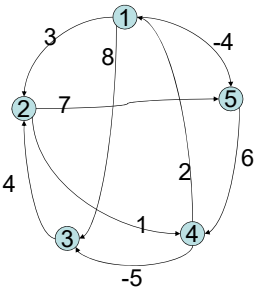


529

Floyd-Warshallの実行例

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

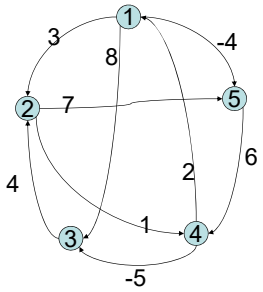


530

Floyd-Warshallの実行例

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

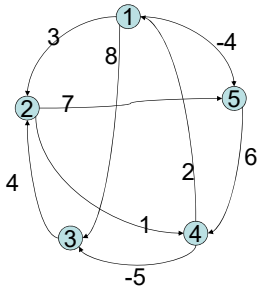


531

Floyd-Warshallの実行例

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

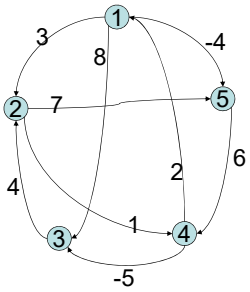
$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$



532

Floyd-Warshallの実行例

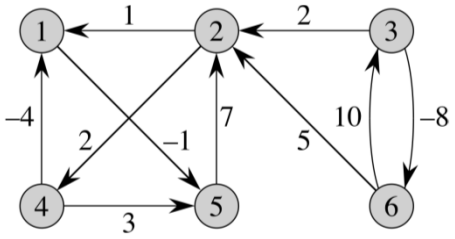
$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$



533

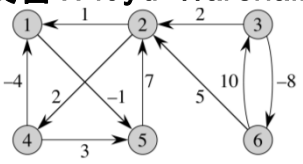
演習：Floyd-Warshall

- 以下のグラフに関して，Floyd-Warshallのアルゴリズムで全点対最短経路を求めよ



534

演習：Floyd-Warshall



$$D^{(6)} = \begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$$

535