

データ構造と アルゴリズムII

第10回
単一始点最短路 (II) / 全点对最短路

495

トポロジカル・ソート順 による緩和

496

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
 - $\Theta(|V| + |E|)$

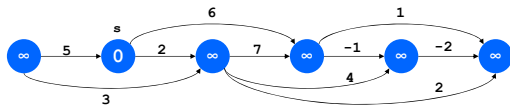
497

方針

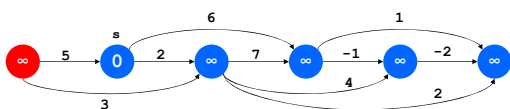
- グラフをトポロジカル・ソートして頂点に線形順序を与える
- ソート順に頂点を選び, その頂点の出辺を緩和する
 - 各頂点は一回だけ選択される

498

アルゴリズムの動き



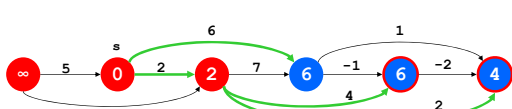
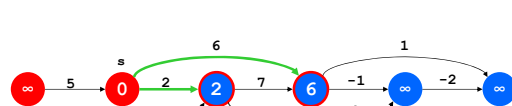
閉路のない有向グラフをトポロジカル・ソートする



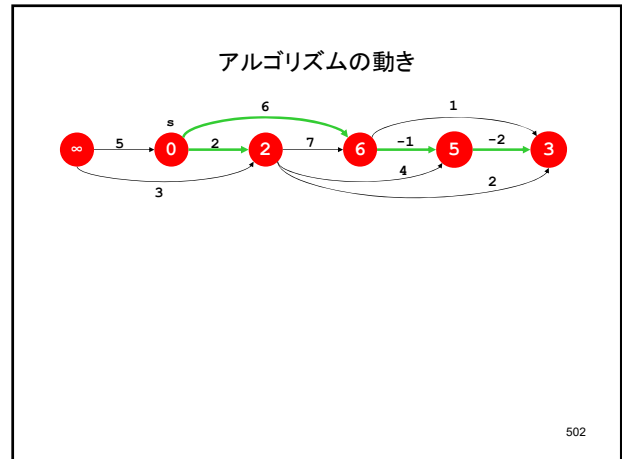
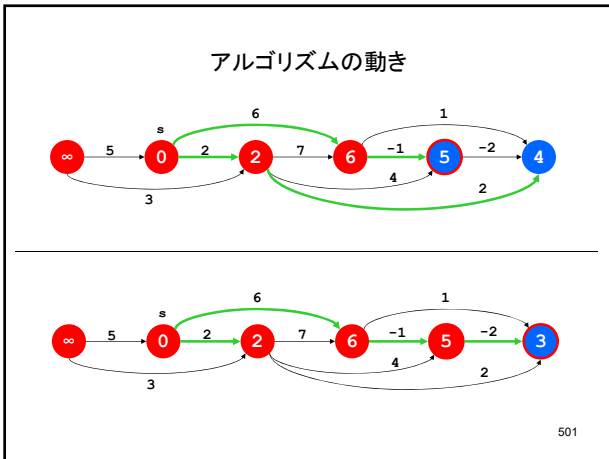
左の頂点から順番に選択, 選択された頂点の出辺を緩和する

499

アルゴリズムの動き



500



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)
    
```

503

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 $d[v] = \delta(s, v)$ 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

504

経路緩和性

$p = \langle v_0, v_1, \dots, v_k \rangle$ が $s = v_0$ から v_k に至る最短路で、 p の辺が $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$ の順序で緩和されたとき、 $d[v_k] = \delta(s, v_k)$ が成立する。

この性質は他の任意の緩和操作とは無関係に成立する。例えば、これらの緩和操作の実行が、その他の任意の緩和操作 (p に関する緩和操作でも良い) の実行とシャッフルされた順序で実行されたとしても、この性質は成立する。

505

計算量

- $\Theta(|V| + |E|)$
 - トポロジカル・ソート $\Theta(|V| + |E|)$
 - 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 $\Theta(1) \dots \Theta(|E|)$

506

ダイクストラの アルゴリズム

507

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経路問題を高速に解く
 - $w(u, v) \geq 0$
- ベルマン・フォードのアルゴリズムより高速に解ける

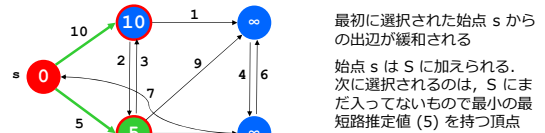
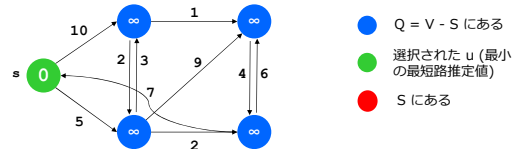
508

ダイクストラ法の方針

- 最小の最短経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経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 $\rightarrow u$ を S に追加 $\rightarrow u$ 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

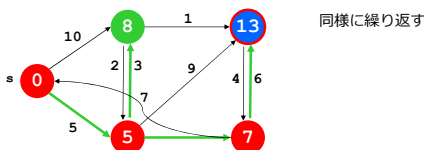
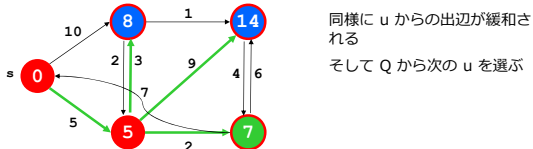
509

ダイクストラのアルゴリズムの動き



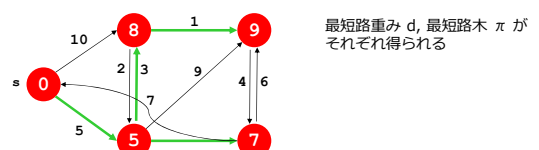
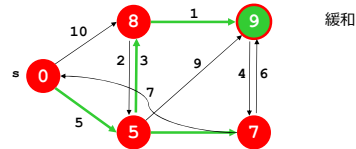
510

ダイクストラのアルゴリズムの動き



511

ダイクストラのアルゴリズムの動き



512

擬似コード

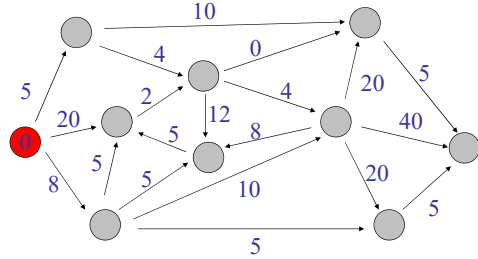
```

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)
    
```

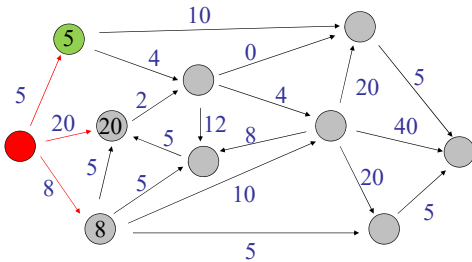
513

演習

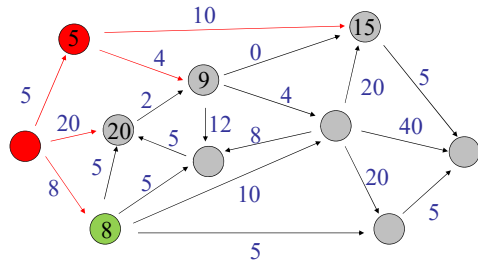
- このネットワークの赤の頂点からの最短距離をダイクストラのアルゴリズムで求めよ



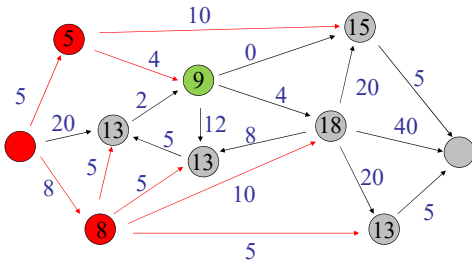
演習



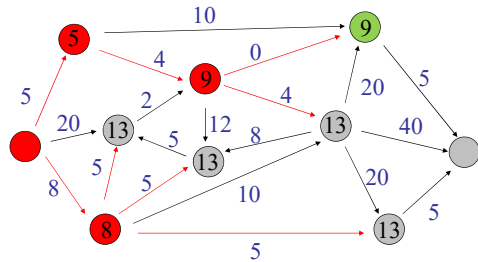
演習



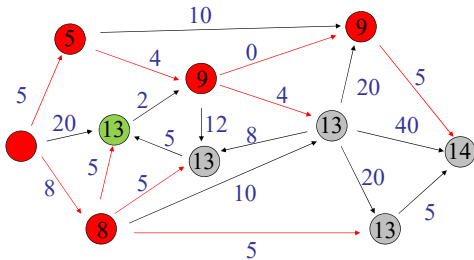
演習



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 $d[v] = \delta(s, v)$ 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開始直前に既に $d[v] = \delta(s, v)$ が各頂点 $v \in S$ について成立している (定理 24.6) ことから証明できる
 - 直感的には, エッジ重みが非負なので, S 中の頂点の値が, これ以上小さく成り得ない
 - 貪欲戦略に基づく選択が正しい

520

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 ... $O(\lg |V|)$
 - DECREASE-KEY ... $O(1)$
 - ダイクストラ法 $O(|V| \lg |V| + |E|)$

521

まとめ

- 単一始点最短路問題を解くアルゴリズム
 - 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

522

差分制約と最短路

523

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
- $Ax \leq b$ 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

$$x_1 - x_2 \leq 5$$

$$x_1 - x_3 \leq 6$$

$$x_2 - x_4 \leq -1$$

$$x_3 - x_4 \leq -2$$

$$x_4 - x_1 \leq -3$$

524

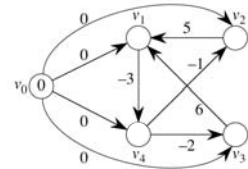
差分制約式系の適用分野

- 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
- 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

525

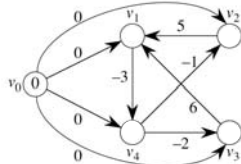
差分制約式系と最短経路 (1)

- 差分制約式系中の各変数 x_1, \dots, x_n に対応する頂点 v_1, \dots, v_n を考える
 - $x_1 - x_2 \leq 5$
 - $x_1 - x_3 \leq 6$
 - $x_2 - x_4 \leq -1$
 - $x_3 - x_4 \leq -2$
 - $x_4 - x_1 \leq -3$
- 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 $x_j - x_i \leq b$ に対して、辺 (v_i, v_j) を導入、重みは b



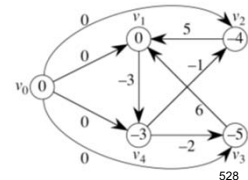
差分制約式系と最短経路 (2)

- この制約グラフの単一始点最短経路を求めたとき、その最短経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
 - $x_1 - x_2 \leq 5$
 - $x_1 - x_3 \leq 6$
 - $x_2 - x_4 \leq -1$
 - $x_3 - x_4 \leq -2$
 - $x_4 - x_1 \leq -3$
- ベルマン・フォードのアルゴリズムが使える $\Rightarrow O(|V| \cdot |E|)$ 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



差分制約式系と最短経路 (3)

- 解答: $(0, -4, -3, -5)$.
 - $x_1 - x_2 \leq 5$
 - $x_1 - x_3 \leq 6$
 - $x_2 - x_4 \leq -1$
 - $x_3 - x_4 \leq -2$
 - $x_4 - x_1 \leq -3$
- 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



25. 全点对最短経路

529

25章の内容

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

530

最短経路の性質

(v_i, \dots, v_k, v_j) が v_i から v_j への最短経路であるとす。すなわち、最短経路は 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:

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

531

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 $i \rightarrow j$ 間の最短経路とする。以下のように帰納的に定義可能。

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

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

532

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 $i \rightarrow j$ 間の最短経路長となる。
- $O(|V|^3 \log |V|)$ 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。

533

全点对最短路アルゴリズム

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

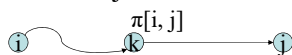
$$W[i, j] = \begin{cases} 0, & i = j \\ w(i, j), & (i, j) \in E \\ \infty, & (i, j) \notin E \end{cases}$$

534

全点对最短路アルゴリズム

出力:

- $n \times n$ の距離行列 $D=(D[i, j])$
 $D[i, j] = \delta(i, j)$
- $n \times n$ の先行点行列 $\pi=(\pi[i, j])$
もし $i \rightarrow j$ の経路がなければ $\pi[i, j]=NIL$,
そうでなければ、 $k=\pi[i, j]$ の時、 $i \rightarrow j$ の最短経路は k から直接 j に至る。



535

```
Extend-Shortest-Paths(D, W)
{ Let D' = (D'[i, j]) be an n x 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: $O(n^3)$

536

```

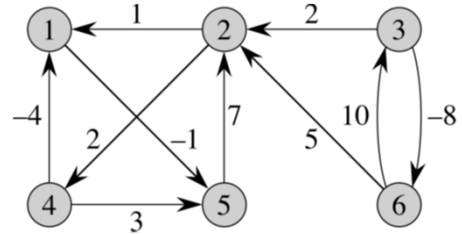
Slow-All-Pairs-Shortest-Paths(G,W)
{ D(1) ← W
  for m=2 to n-1 do
    D(m) ← Extend-Shortest-Paths(D(m-1), W)
  return D(n-1)
}
    
```

Time Complexity: $O(n^4)$

537

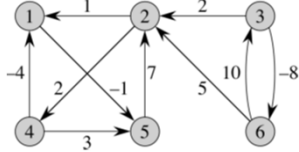
演習: 全点对最短路

- 動的計画法で以下のグラフの全点对最短路を求めよ



538

演習: 全点对最短路



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

539

演習: 全点对最短路

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

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

540

演習: 全点对最短路

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

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

541

演習: 全点对最短路

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

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

542

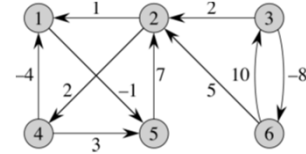
演習: 全点对最短路

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

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

543

演習: 全点对最短路



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

544

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-Shortest-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(経路する頂点数が 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.

545

```
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: $O(n^3 \log n)$

546

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. $i \rightarrow j$ 間の経路が (i, u_1, \dots, u_m, j) とすると, u_1, \dots, u_m が中間頂点.

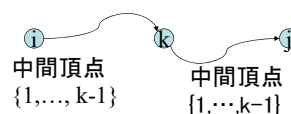
547

Floyd-Warshallアルゴリズム

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

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

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



548

Floyd-Warshallアルゴリズム

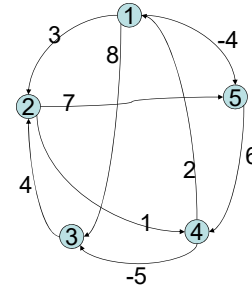
```

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: $O(n^3)$

549

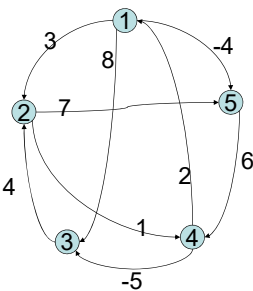
Floyd-Warshallの実行例



550

Floyd-Warshallの実行例

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

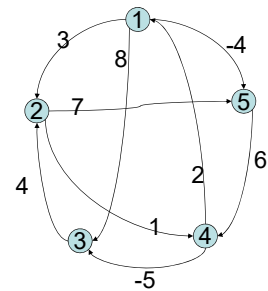


551

Floyd-Warshallの実行例

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

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

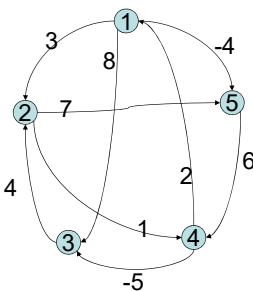


552

Floyd-Warshallの実行例

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

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

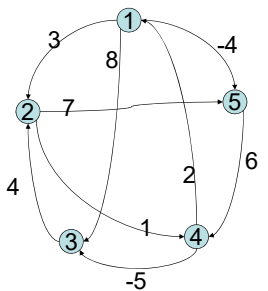


553

Floyd-Warshallの実行例

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

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



554

Floyd-Warshallの実行例

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

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

555

Floyd-Warshallの実行例

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

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

556

Floyd-Warshallの実行例

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

557

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对最短経路を求めよ

558

演習 : Floyd-Warshall

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

559