最短路問題

出発地から目的地までの最適な経路を求める問題は，ネットワーク上の最短路問題として知られている．この問題は線形計画問題の一つであるが，その特殊な問題構造から一般の線形計画問題よりも易しく，シンプレックス法とは異なる専用のアルゴリズムがある．

最短路問題

GIVEN ネットワーク
- 有向グラフ $G = (V, E)$
- 各枝 $(i, j) \in E$ の長さ c_{ij}

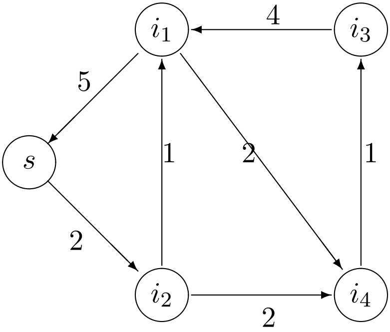
FIND 特定の頂点 s から他のすべての頂点への最短路

＊　頂点 i から頂点 j の有向路の中で，その有向路に含まれる枝の長さの和が最小となるものを i から j への［　　　］という．

＊　特定の頂点からの最短路を求める問題を［　　　　　　］ということもある．

仮定 G には s から各頂点への有向路が存在する
(s から各頂点 i への枝 (s, i) を追加し，この枝の長さを非常に大きな値とすれば，最短路は変わらずに必ず仮定を満たす)

最短路問題

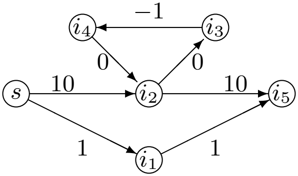


負閉路

枝の長さの総和が負である有向閉路を $\boxed{}$ という．

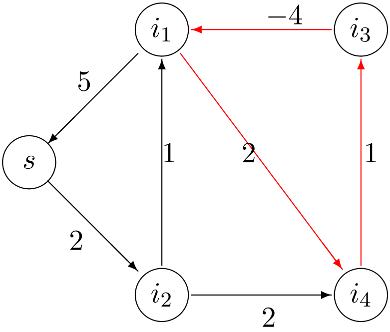
負閉路のある最短路問題

下図に示す問題例で s から i_5 への最短路を考えよう．



約束 s から頂点 i への有向路の途中に負閉路がある場合，最短路問題ではこの負閉路を出力する

負閉路のある最短路問題



最短路の性質 1

補題

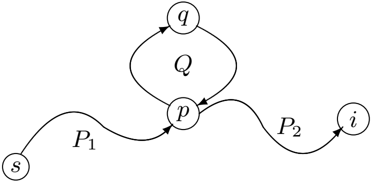
頂点 s から i への最短路が存在すれば，最短路の中に，s から i への初等的な有向路（有向道）が存在する．

s から i への有向路の途中に負閉路が存在しない $\iff$ s から i への最短路が存在する

$\therefore$ s から i への最短路が初等的でない（途中に閉路 Q が存在）$\Rightarrow$ 閉路 Q の長さは $\boxed{}$

s から i への有向路 P に閉路が存在したとき $\boxed{}$

- P_1:s から p への有向路
- Q:p, q を通る閉路
- P_2:p から i への有向路



$P' = \boxed{}$ も s から i への有向路

$\boxed{}$

最短路の性質 2

定理

頂点 s から i へのある有向路の長さを $d(i)$ で表す．ただし，$d(s) = 0$ とする．すべての頂点 i で $d(i)$ が最短路長であることの必要十分条件は，

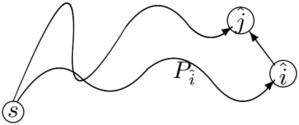
$$d(j) \leq d(i) + c_{ij}, \quad \forall (i,j) \in E \qquad (\boxed{})$$

が成り立つことである

[必要条件]　(対偶) $\boxed{}$ である枝 $(\hat{i}, \hat{j})$ があるとき

$\Rightarrow$ $d(\hat{i})$ を達成する s から $\hat{i}$ への有向路 $P_{\hat{i}}$ に枝 $(\hat{i}, \hat{j})$ を加えてできる，s から $\hat{j}$ への有向路の長さは $d(\hat{j})$ よりも $\boxed{}$

$\Rightarrow$ $d(\hat{j})$ は最短路長ではない．



最短路の性質 2

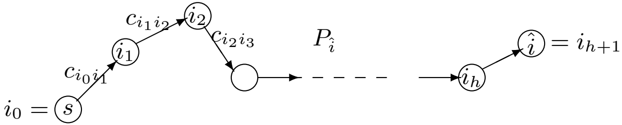
定理

頂点 s から i へのある有向路の長さを $d(i)$ で表す．ただし，$d(s)=0$ とする．すべての頂点 i で $d(i)$ が最短路長であることの必要十分条件は，

$$d(j) \le d(i) + c_{ij}, \quad \forall (i,j) \in E$$

が成り立つことである

[十分条件] s から頂点 $\hat{i}$ への任意の有向路 $P_{\hat{i}} = \langle s, i_1, i_2, \ldots, i_h, \hat{i} \rangle$ の長さを考える．



便宜上 $i_0 = s$，$i_{h+1} = \hat{i}$ とする．

∴ $d(\hat{i})$ が最短路長となる．

最短路の性質 3

系

頂点 s から各頂点 i への最短路長を $d(i)$ で表す．s から $\hat{i}$ への有向路 $P_{\hat{i}}$ が s から $\hat{i}$ への最短路であるための必要十分条件は，

$$d(j) = d(i) + c_{ij}, \quad \forall (i,j) \in P_{\hat{i}}$$

が成り立つことである．

s から $\hat{i}$ への有向路 $P_{\hat{i}} = \langle s, i_1, i_2, \ldots, i_h, \hat{i} \rangle$ が最短路であるとき，

$$P_{\hat{i}} \text{ の長さ}$$

$$= \sum_{(i,j)\in P_{\hat{i}}} c_{ij} = \sum_{k=0}^{h} c_{i_k i_{k+1}} \leq \sum_{k=0}^{h} (d(i_{k+1}) - d(i_k)) = d(\hat{i}) - d(s) = d(\hat{i}) = P_{\hat{i}} \text{ の長さ}$$

満たす．

最短路の性質 4

系

有向路 $P_{\hat{i}} = \langle s, i_1, i_2, \ldots, i_h, \hat{i} \rangle$ **が** s **から** $\hat{i}$ **への最短路であるとき，各** $\ell = 1, 2, \ldots, h$ **に対して，** $P_{\hat{i}}$ **の部分有向路** $P_{i_\ell} = \langle s, i_1, i_2, \ldots, i_\ell \rangle$ **は** s **から** i_ℓ **への最短路である．**

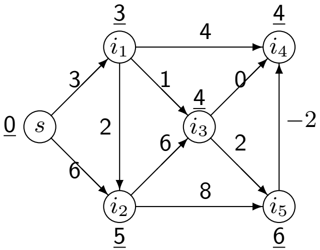
最短路長と最短路

下図の最短路問題の問題例

- **各枝** (i,j) **上の数字がその枝の長さ** c_{ij}
- **各頂点** i **のそばに** s **からその頂点への最短路長** $d(i)$

を示す．

1. **どの枝でも距離に関する三角不等式** $(d(j) \le d(i) + c_{ij}, \quad \forall (i,j) \in E\ )$ **が成り立っていることを確認しよう**
2. **最短路上の枝** (i,j) **では** $d(j) = d(i) + c_{ij}$ **が成り立っていることを確認しよう (例えば，**s **から** i_4 **への最短路** $\langle s, i_1, i_3, i_4 \rangle$**)**
3. **最短路の部分有向道が最短路となることを確認しよう**



最短路問題の LP 表現

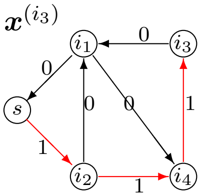
変数： $\boldsymbol{x} \in \mathbb{R}_+^E$ **（各枝に対応する非負変数）**

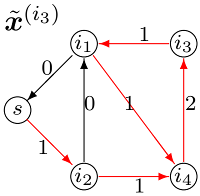
- $x_{ij} = \displaystyle\sum_{\hat{i} \in V \setminus \{s\}} x_{ij}^{(\hat{i})}$
- $x_{ij}^{(\hat{i})}$: s **から頂点** $\hat{i}$ **への有向路** $P_{\hat{i}}$ **に枝** (i,j) **が含まれる回数**

$\boldsymbol{x}^{(\hat{i})} \in \mathbb{R}^E$ **は**

$$\sum_{j:(i,j)\in E} x_{ij}^{(\hat{i})} - \sum_{k:(k,i)\in E} x_{ki}^{(\hat{i})} = \begin{cases} 1, & i = s \\ -1, & i = \hat{i} \\ 0, & i \in V \setminus \{s, \hat{i}\} \end{cases}$$

を満たす





LP で定式化

有向路に含まれる回数で定義される x は (SP) の実行可能解である

$$
(\mathrm{SP})\left|\begin{array}{ll}
\underset{\boldsymbol{x}}{\textbf{最小化}} & \displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} \\
\textbf{条　件} & \displaystyle\sum_{j:(i,j)\in E} x_{ij} - \sum_{k:(k,i)\in E} x_{ki} = \begin{cases} n-1, & i = s \\ -1, & i \in V \setminus \{s\} \end{cases} \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{array}\right.
$$

(SP) **の最適解 x^* から最短路が得られる**?

- x^*_{ij} **が整数だと最短路に含まれる回数を表す**

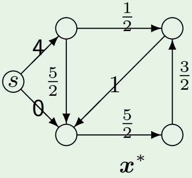
  有向路に含まれる回数で与えられる SP の実行可能解よりも，目的関数値が小さくなることより，この有向路が最短路を成す

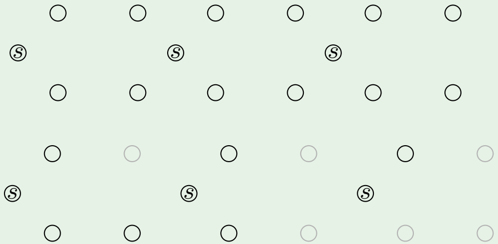
x^* を有向路に分解

- $\{(i,j) \in E \mid x^*_{ij} > 0\}$ からなる部分グラフを有向路に分解
- それぞれの有向路に各枝が含まれる回数を正の定数倍したものを足し合わせると x^* となるようにできる

s から $\{(i,j)|x^*_{ij} > 0\}$ の枝を行けるところまで辿り，s からの有向路 P を作る．この有向路上で x^*_{ij} の値が最も小さい値 α でこの有向路に各枝が含まれる回数を定数倍する．次に，有向路 P に含まれる各枝 (i,j) の x^*_{ij} の値から α を引いたものを新たに x^* として，同じ操作を繰り返す．

有向路への分解





有向路への分解後

- 頂点 $\hat{i}$ へ 2 本の有向路 $P_{\hat{i}}$, $P'_{\hat{i}}$
  - $P_{\hat{i}}$ に含まれる回数を $\lambda(>0)$ 倍
  - $P'_{\hat{i}}$ に含まれる回数を $\mu(>0)$ 倍

  したと仮定
- $P_{\hat{i}}$, $P'_{\hat{i}}$ に関する目的関数での値は，$\lambda \sum_{(i,j)\in P_{\hat{i}}} c_{ij} + \mu \sum_{(i,j)\in P'_{\hat{i}}} c_{ij}$
- $P_{\hat{i}}$, $P'_{\hat{i}}$ のうちの 1 本のみを用いても，目的関数は増加しない

$$\lambda \sum_{(i,j)\in P_{\hat{i}}} c_{ij} + \mu \sum_{(i,j)\in P'_{\hat{i}}} c_{ij} \geq (\lambda+\mu)\min\{\sum_{(i,j)\in P_{\hat{i}}} c_{ij}, \sum_{(i,j)\in P'_{\hat{i}}} c_{ij}\}$$

$\Rightarrow$ x^* がある頂点への複数本の有向路に分解されたときに，これらの有向路の 1 本のみを使うようにすることを繰り返すことで，s から各頂点への有向路が 1 本となるような (SP) の最適解を得ることができる

定理

線形計画問題 (SP) に最適解があるとき，整数の最適解が必ず存在し，s から各頂点への最短路に対応する．

(SP) の双対問題と相補スラック性条件

制約：

$$\begin{array}{ccccc} & \begin{array}{c} A \\ \text{枝} \\ (i,j) \end{array} & x & & b \\ \begin{array}{c} \text{頂} \\ \text{点} \\ i \end{array} & \begin{bmatrix} \\ \\ \end{bmatrix} & \begin{bmatrix} \\ x_{ij} \\ \\ \end{bmatrix} & = & \begin{bmatrix} n-1 \\ -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \end{array}$$

Exercise

双対問題，相補スラック性条件を書こう

最短路問題に対するアルゴリズム

1. Dijkstra（ダイクストラ）法
2. Bellman-Ford（ベルマン-フォード) 法

Dijkstra 法準備

- s から各頂点 i への有向路の長さ $d(i)$: $\boxed{}$
- 距離に関する三角不等式 $\boxed{}$ を満たす距離ラベル $d(i)$ を求める
- 基本操作：頂点 s から扇状に探索の木を広げていき，s から近い順に各頂点 i の距離ラベルを決める
  - $\boxed{}$:最短路長と等しくなりその後ラベルの更新がされない距離ラベル

    V_P 永久ラベルをもつ頂点の集合
  - $\boxed{}$:仮の最短路長の値でその後に更新の可能性がある距離ラベル

    V_T 一時ラベルをもつ頂点の集合

仮定 各枝の長さは非負 $(c_{ij} \geq 0, \quad \forall (i,j) \in E)$

Dijkstra 法

$V_P \leftarrow \emptyset$, $V_T \leftarrow V$

$$d(i) \leftarrow \begin{cases} 0 & i = s \\ \infty & i \in V \setminus \{s\} \end{cases}, \ \mathrm{pred}(i) \leftarrow \begin{cases} s & i = s \\ \emptyset & i \in V \setminus \{s\} \end{cases}$$

while $V_P \neq V$ **do**

  $\min\{d(i) \mid i \in V_T\}$ **を達成する頂点** $\tilde{i}$ **を選ぶ**

  $V_P \leftarrow V_P \cup \{\tilde{i}\}$, $V_T \leftarrow V_T \setminus \{\tilde{i}\}$

  for $j \in V_T$with $(\tilde{i}, j) \in E$ **do**

    if $d(j) > d(\tilde{i}) + c_{\tilde{i}j}$ **then**

      $d(j) \leftarrow d(\tilde{i}) + c_{\tilde{i}j}$, $\mathrm{pred}(j) \leftarrow \tilde{i}$

    end if

  end for

end while

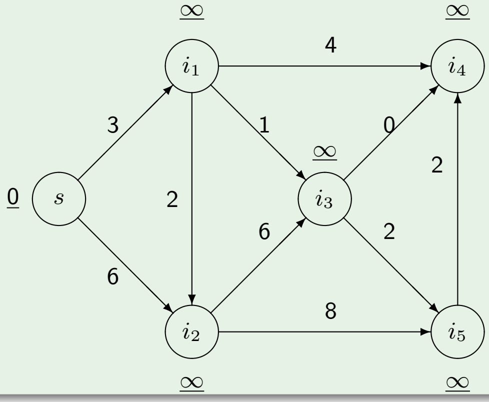
初期化 $V_P = \emptyset$，$V_T = V$ として，頂点 s には一時ラベル $d(s) = 0$ を貼り，s 以外の頂点 i には $d(i) = \infty$ を貼る．

繰り返し 一時ラベルをもつ頂点集合 V_T の中から $d(i)$ の最も小さい頂点 $\tilde{i}$ を選んで，$\tilde{i}$ の距離ラベル $d(\tilde{i})$ を永久ラベルとする．$\tilde{i}$ を始点とする各枝 $(\tilde{i}, j) \in E$ で $d(j) \leq d(\tilde{i}) + c_{\tilde{i}j}$ を満たすように終点 j の距離ラベル $d(j)$ を更新する．

各頂点 i に対する pred(i) は，s から i への長さ $d(i)$ の有向路で頂点 i の直前に訪れる頂点を指す．

Dijkstra 法の例

- 枝 (i,j) 上の数字はその枝の長さ c_{ij}
- 頂点そばの下線付き数字が距離ラベル
- 二重線の頂点がその繰り返しで $\tilde{i}$ として選ばれた頂点
- 塗りつぶしてある頂点は永久ラベル
- 太線は $\{(\mathrm{pred}(i), i) \in E \mid i \in V\}$



Dijkstra 法の正当性 1

Proposition

♠ 各頂点 i の有限値をとる距離ラベル $d(i)$ は，集合 V_P の頂点だけを通って s から i へ向かう有向路の中で最短の長さを表す

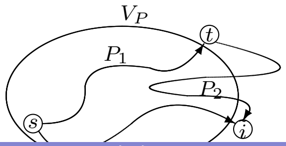
- $G_{(V_P)}$: G から両端点が共に V_T にある枝を除いたグラフ
- $G_{(V_P)}$ で $d(i)$ が距離に関する三角不等式を満たすことを示す．
  - 帰納法
  - 頂点 $\tilde{i}$ を V_P に入れる前に成立
  - $\tilde{i}$ を加えた後，$\tilde{i}$ を始点とする各枝 $(\tilde{i}, j) \in E$ の終点の距離ラベル $d(j)$ を更新するので成立

Dijkstra 法の正当性 2

Proposition

V_T の中で，距離ラベル $d(i)$ の最も小さい頂点を永久ラベルとできる

- 帰納法
- ある繰り返しで，V_P の頂点の距離ラベルはすべて最短路長となっていたとする．
- $d(i)$ の最も小さい頂点 $\tilde{i} \in V_T$
- s から $\tilde{i}$ への有向路 P
  - P が V_P の頂点のみを通る場合，
  - P が途中で V_T を通る場合，
    - ★
    - ★ P_1:
    - ★ P_2:
  - P の長さ $=$ P_1 の長さ $+$ P_2 の長さ $\geq$ P_1 の長さ $\geq d(t) \geq d(\tilde{i})$



Dijkstra 法

$V_P \leftarrow \emptyset$, $V_T \leftarrow V$

$$d(i) \leftarrow \begin{cases} 0 & i = s \\ \infty & i \in V \setminus \{s\} \end{cases}, \ \mathrm{pred}(i) \leftarrow \begin{cases} s & i = s \\ \emptyset & i \in V \setminus \{s\} \end{cases}$$

while $V_P \neq V$ **do**
  $\min\{d(i) \mid i \in V_T\}$ **を達成する頂点** $\tilde{i}$ **を選ぶ**
  $V_P \leftarrow V_P \cup \{\tilde{i}\}$, $V_T \leftarrow V_T \setminus \{\tilde{i}\}$
  for $j \in V_T$with $(\tilde{i}, j) \in E$ **do**
    if $d(j) > d(\tilde{i}) + c_{\tilde{i}j}$ **then**
      $d(j) \leftarrow d(\tilde{i}) + c_{\tilde{i}j}$, $\mathrm{pred}(j) \leftarrow \tilde{i}$
    end if
  end for
end while

アルゴリズムが終了したとき，pred(i) **を順に辿ることで，最短路が得られる．**

ダイクストラ法は繰り返しごとに 1 **つの頂点に永久ラベルを貼り，距離ラベルを確定することより，**$\boxed{}$**とも呼ばれる．**

Bellman-Ford 法準備

負の枝長も許す

- 距離に関する三角不等式 $\boxed{}$ を満たす距離ラベル $d(i)$ を求める．
- Dijkstra 法とは異なり，一度選択した頂点の距離ラベルの値が $\boxed{} \Rightarrow \boxed{}$
- V_T:頂点 i を始点とする枝 (i, j) で $d(j) \leq d(i) + c_{ij}$ を満たさない可能性がある i の集合
- 基本操作：V_T から適当に 1 頂点 $\tilde{i}$ を選び，$\tilde{i}$ を始点とする各枝 $(\tilde{i}, j) \in E$ で

  $$d(j) \leq d(\tilde{i}) + c_{\tilde{i}j}$$

  を満たすように終点 j の距離ラベル $d(j)$ を更新する
  - 初期化では，頂点 s の距離ラベルを $d(s) = 0$ とし，s 以外の頂点 i では $d(i) = \infty$，$V_T = \boxed{}$ とできる．
  - 距離ラベルの更新後は V_T から $\tilde{i}$ を除くことができるが，一方で，$\boxed{}$ を V_T に加える
  - $V_T = \emptyset$ となったとき終了

Bellman-Ford 法

$$d(i) \leftarrow \begin{cases} 0 & i = s \\ \infty & i \in V \setminus \{s\} \end{cases}, \ \mathrm{pred}(i) \leftarrow \begin{cases} s & i = s \\ \emptyset & i \in V \setminus \{s\} \end{cases}$$

$V_T \leftarrow \{s\}$
while $V_T \neq \emptyset$ **do**
  $\tilde{i} \in V_T$ **を選ぶ**
  $V_T \leftarrow V_T \setminus \{\tilde{i}\}$
  for $j \in V$ with $(\tilde{i}, j) \in E$ **do**
    if $d(j) > d(\tilde{i}) + c_{\tilde{i}j}$ **then**
      $d(j) \leftarrow d(\tilde{i}) + c_{\tilde{i}j}$, $\mathrm{pred}(j) \leftarrow \tilde{i}$, $V_T \leftarrow V_T \cup \{j\}$
    end if
  end for
end while

Bellman-Ford 法の有限性１

Proposition

各頂点への最短路が存在するときは，各頂点が V_T に加えられる回数が $\boxed{\qquad\qquad}$ でありアルゴリズムは必ず終了する．

$\therefore$

- $d(i)$ が有限値をとるとき，$d(i)$ は s から i への $\boxed{\qquad\qquad\qquad\qquad}$
- 頂点 i が V_T に加わるとき，$d(i)$ は必ず $\boxed{\qquad\qquad}$ している
- 距離ラベル $d(i)$ が $\boxed{\qquad\qquad}$ に等しくなった後は，ラベル値は変更されない
  $\Rightarrow d(i)$ が最短路長に等しくなった頂点が V_T から選択されると，その後は V_T に加わることがない

Bellman-Ford 法の有限性 2

$\boxed{}$ が存在するときは，ある頂点 i の距離ラベル $d(i)$ を無限に小さくできるためにアルゴリズムが終了しない．

解決法 V_T を $\boxed{}$ で持たせて，V_T に格納された順に頂点 $\tilde{i}$ を選択する

キューをわかりやすくするために，$V_T^0, V_T^1, V_T^2, \ldots$ を準備

- $V_T^0 = \boxed{}$
- V_T^{k-1} にある頂点 $\tilde{i}$ が選択され，枝 $(\tilde{i}, j) \in E$ の終点 j の距離ラベルが更新されて新たに V_T に加えられるときに，V_T^k に加えるとする．

Proposition

V_T^k から頂点 i が選択されると，$d(i)$ は s から i までの $\boxed{}$ 有向路の中での最短の長さ以下になっている．

- 最短路が存在するときは，初等的な最短路が存在する．
- 初等的な有向路の枝数は高々(頂点数 $-\ 1$) である
- 頂点数を n としたとき，V_T^n に格納された頂点があるときは，この頂点を含む負閉路がある．

Bellman-Ford 法 (修正)

$d(i) \leftarrow \begin{cases} 0 & i = s \\ \infty & i \in V \setminus \{s\} \end{cases}$, $\mathrm{pred}(i) \leftarrow \begin{cases} s & i = s \\ \emptyset & i \in V \setminus \{s\} \end{cases}$

$k \leftarrow 0$, $V_T^0 \leftarrow \{s\}$, $V_T^\ell \leftarrow \emptyset \ (\ell = 1, \ldots, n)$

while $k < n$ **do**

  while $V_T^k \neq \emptyset$ **do**

    $\tilde{i} \in V_T^k$ **を選ぶ**

    $V_T^k \leftarrow V_T^k \setminus \{\tilde{i}\}$

    for $j \in V$ with $(\tilde{i}, j) \in E$ **do**

      if $d(j) > d(\tilde{i}) + c_{\tilde{i}j}$ **then**

        $d(j) \leftarrow d(\tilde{i}) + c_{\tilde{i}j}$, $\mathrm{pred}(j) \leftarrow \tilde{i}$

        if $j \notin V_T^k$ **then**

          $V_T^{k+1} \leftarrow V_T^{k+1} \cup \{j\}$

        end if

      end if

    end for

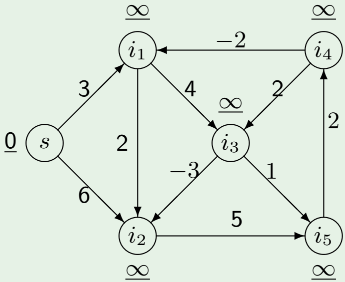
  end while

  $k \leftarrow k + 1$

end while

Bellman-Ford 法の例

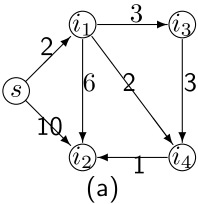
- 枝 (i, j) 上の数字はその枝の長さ c_{ij}
- 頂点そばの下線付き数字が距離ラベル
- 二重線の頂点がその繰り返しで $\tilde{i}$ として選ばれた頂点
- 太線は $\{(\mathrm{pred}(i), i) \in E \mid i \in V\}$

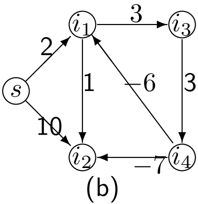


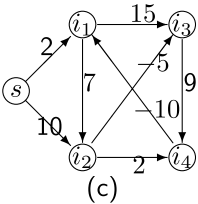
$$V_T^0 = \{s\}$$

演習問題

1. **下図** (a)(b) **の問題で始点** s **から各頂点への最短路を求めよ．** (P90)







2. **下図において，始点** s **から各頂点への最短路を求めよ．**