「基礎OR」／「OR演習」
第5回

10/31/2017

中間試験のお知らせ

- 中間試験日程
  2017／11／14(火)
- 時間：　13:00～14:30(3限)
- 場所(変更ありえます)
  - 56－103, (56－102)教室（座席指定制）
    - 001～070まで
    - 071～（過年度を含む）
- 11／14の4限はこれまでの講義の補足
- 招聘講師による特別講義(出席をとります)
- 所健一先生(電力中央研究所、博士(工学))

主なポイント

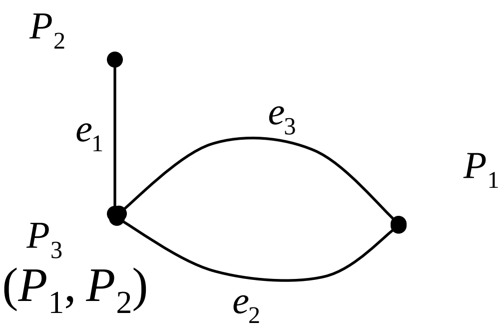
- 線形計画法
  - 単体法で解けること
  - 最適性、感度分析、基底形式、基底解など用語
- 双対問題
  - ラグランジュ緩和による導出
  - 相補スラック条件の説明
- DEA
  - 入力―出力表からLP定式化、双対問題
  - ソルバー結果（与えられている）から効率的か
  - 効率的でない場合の目標を結果から読み記述
- ネットワーク最適化
- 整数計画法、組合せ最適化

ネットワーク上の最適化（ネットワーク計画）の代表的問題

グラフの定義

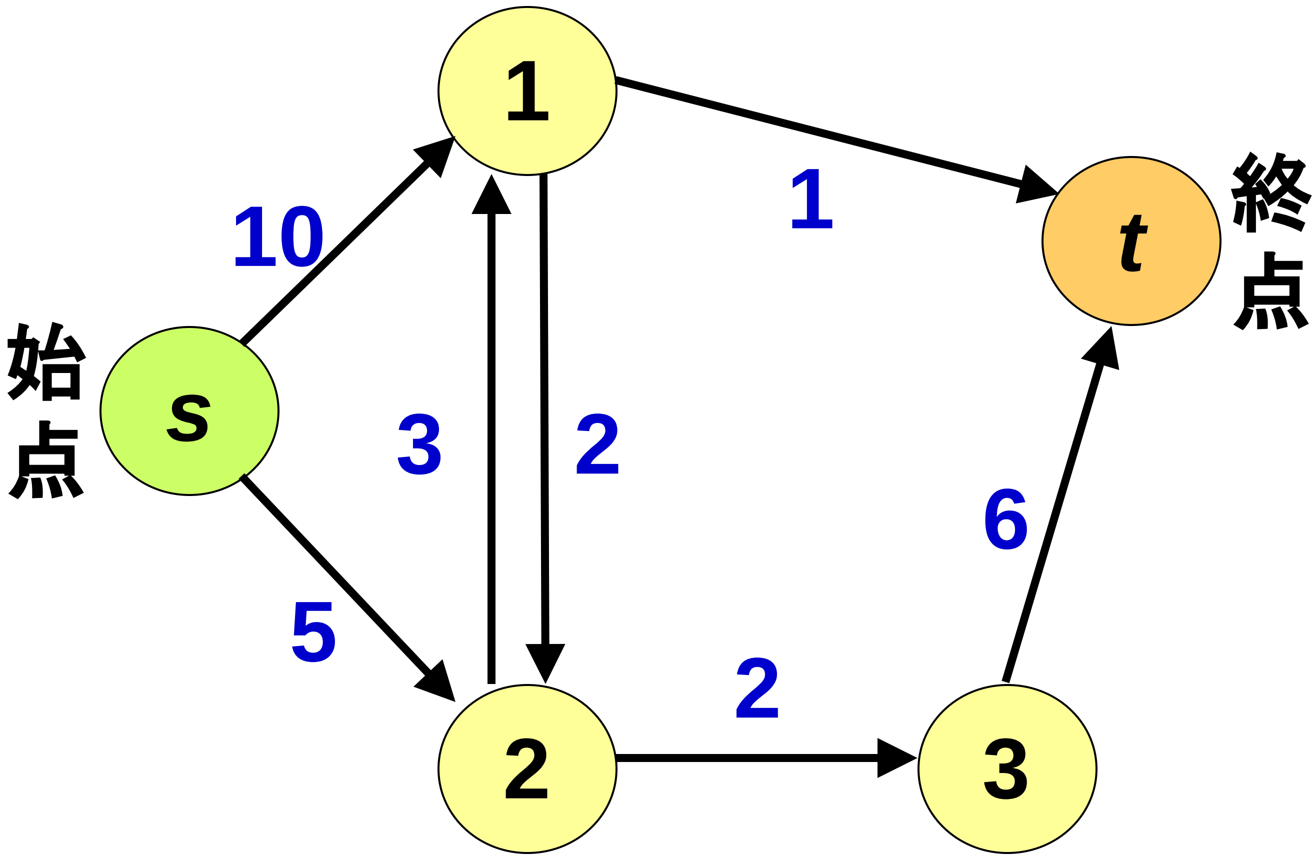
定義 集合$V(\neq\varphi)$と、$V\times V$(非順序対で考える*)の元と対応をもつ集合Eの組 $G=(V,E)$を（無向）グラフといい、Vの要素を点または頂点、Eの要素を辺という。辺に方向を考えるとき、有向グラフとよぶ。
*)順序対で考える場合は点と点の順番を考慮するため、有向グラフになる。

$G_1=(V_1,E_1)$,
$V_1=\{P_1, P_2, P_3\}$,
$E_1=\{e_1, e_2, e_3\}$,
$e_1=(P_2, P_3)$, $e_2=(P_1, P_2)$,$e_3=(P_1, P_2)$



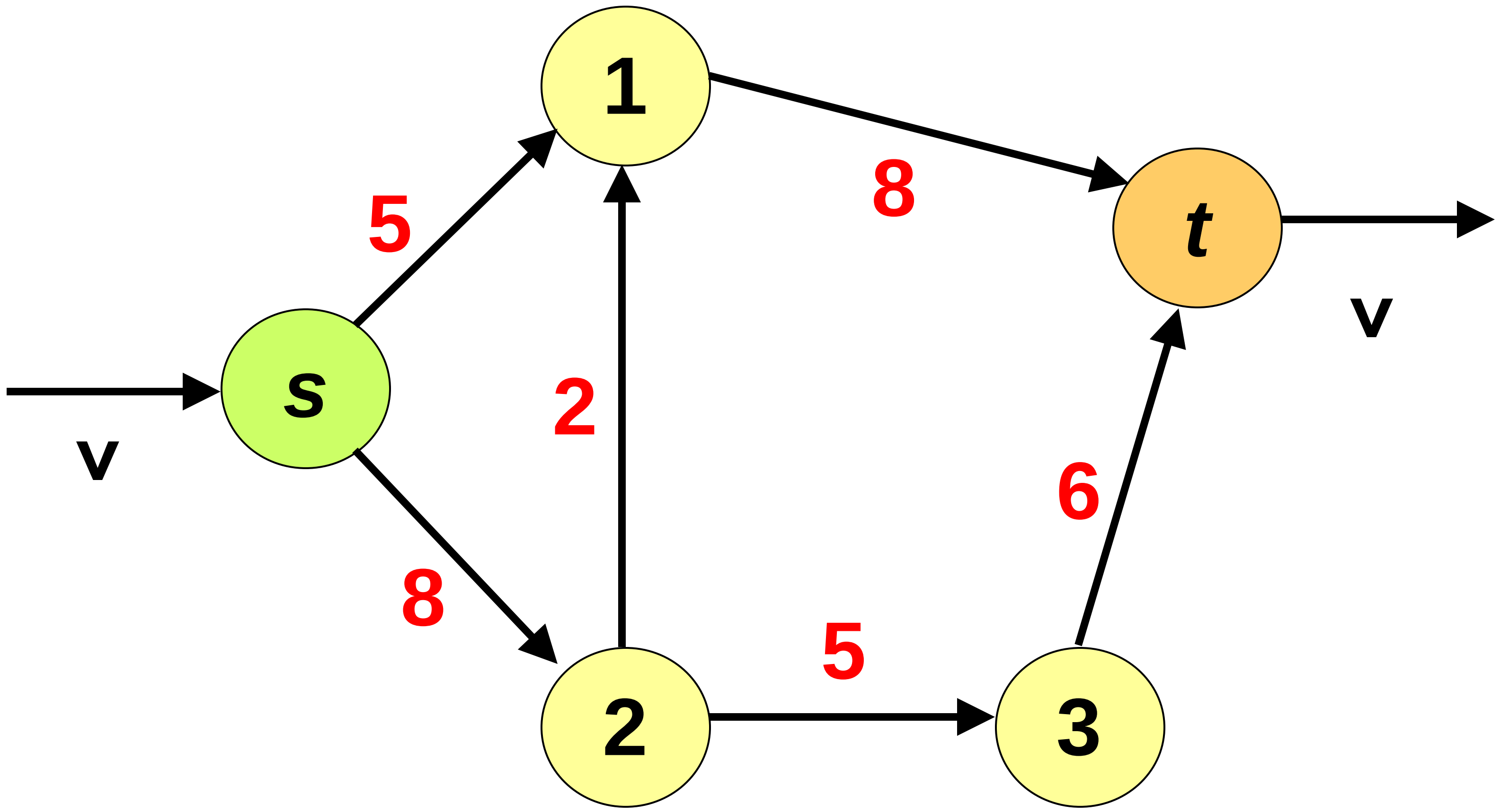
- 辺eに対し、$e=(P,Q)$のとき、「P,Qは辺eの端点である」「P,Qは辺eに接続する」「P,Qは隣接する」「辺eはPまたはQに接続する」という。また辺e_1, e_2が同じ点Pに接続するとき、e_1とe_2は隣接するという。
- 同じ端点をもつ辺が複数あるとき、それらを多重辺といい、端点が同じ点をループという。多重辺やループをもたないグラフを単純グラフ、単純グラフでないグラフを多重グラフという。$V\times V$の元とEの要素との対応は1対1ではない（多重辺が存在する場合）。

最短路問題



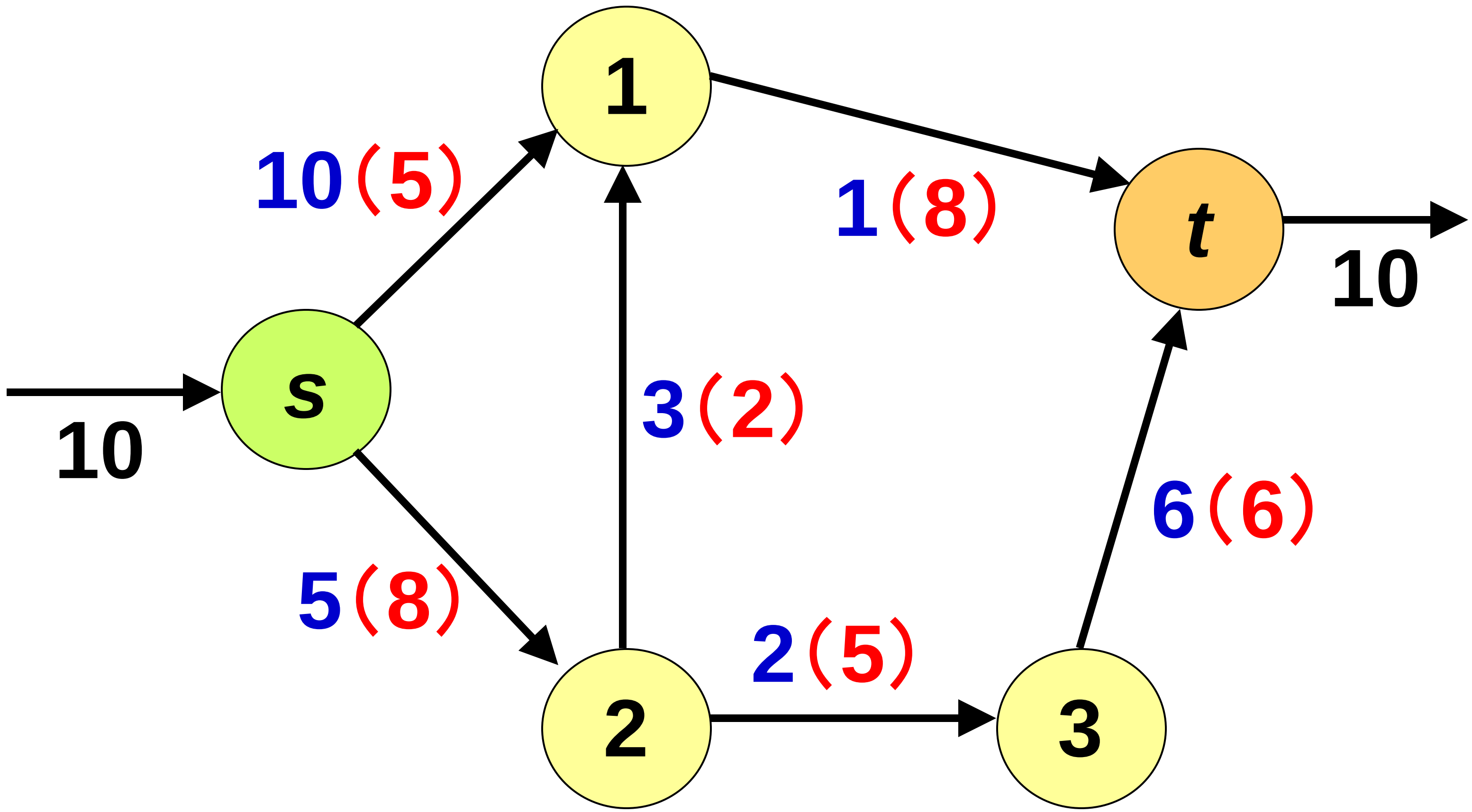
各枝の数字は、枝の距離(費用)を表す

最大流問題



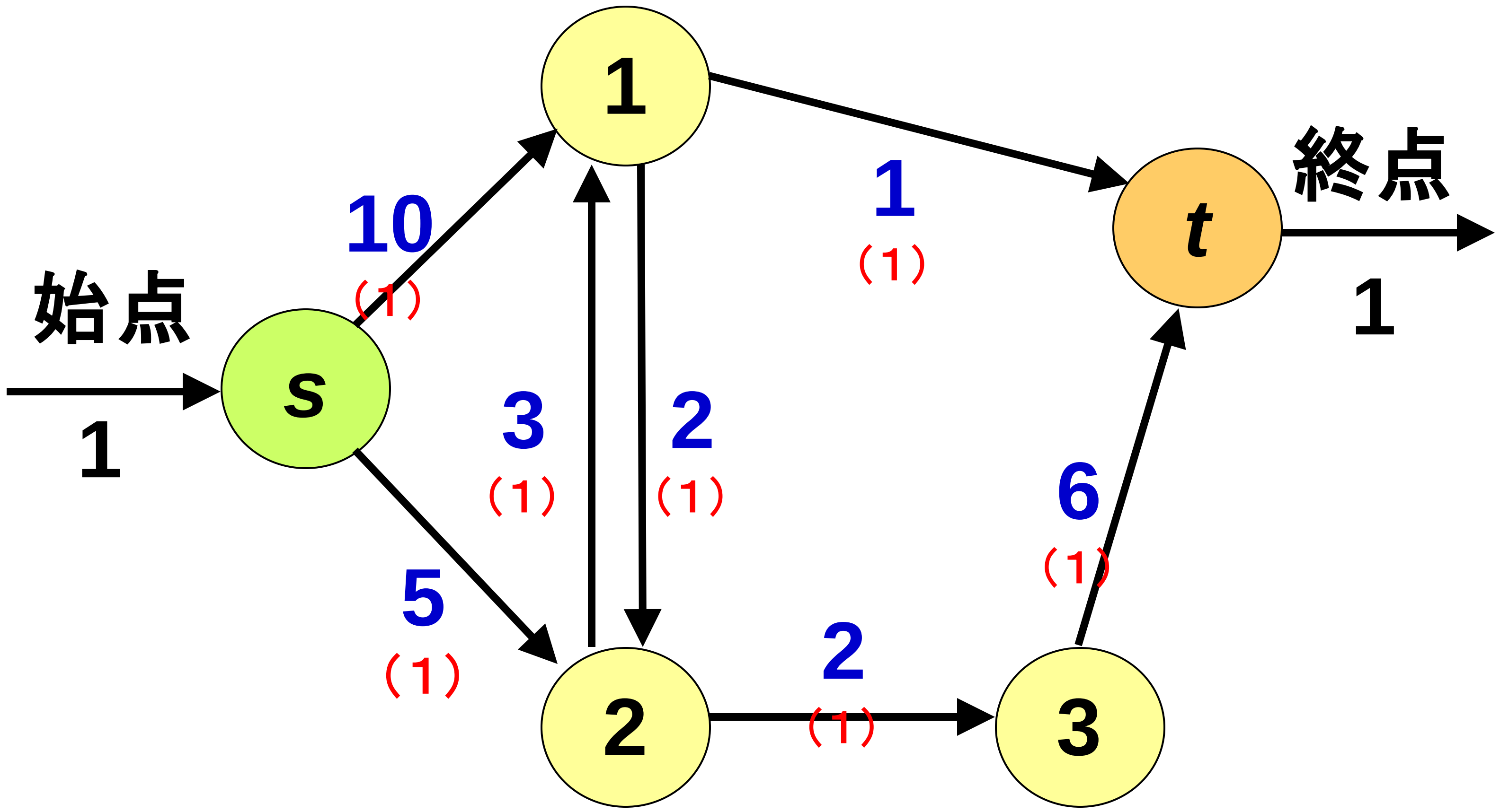
各枝の数字は、枝の容量を表す

最小費用流問題



各枝の数字は、各点間の単位当たり費用、
（　）内の数字は、各枝の容量を表す

最短路問題
（＝流量1の最小費用流問題）



各枝の数字は、各点間の距離（費用）、
（　）内の数字は、各枝の容量を表す

木(tree)の定義

定義 閉路をもたない連結グラフを木(tree)グラフという。

定理　2個以上の頂点を持つグラフTにおいて、次の命題は同値である。

(i) Tは木グラフである。

(ii) Tの任意の2頂点に対し、それらを結ぶただ一つの道が存在する。

(iii) Tは連結であり、かつTのどの辺を除いても連結ではなくなる。

(iv) Tは閉路を含まず、かつ辺をどのように1本加えても閉路を1つもつグラフとなる。

定理 n個の頂点からなる連結グラフが木グラフであるための必要十分条件はn-1個の辺をもつことである。

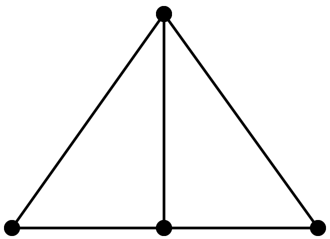
全域木(spanning tree)

定義 グラフGの部分グラフ*T*がGの全ての頂点を含む木グラフであるとき、*T*をGの全域木という。

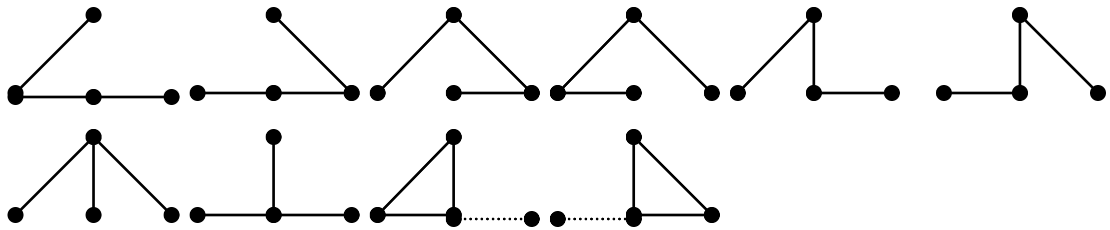
- 次のグラフの全域木を全て求めよ。

  頂点の数は4つなので、全域木は3つの辺をもつ。

  対象とするグラフは5つの辺を持つため、全域木の辺となる3つの辺の選び方は${}_5C_3=10$通りである。

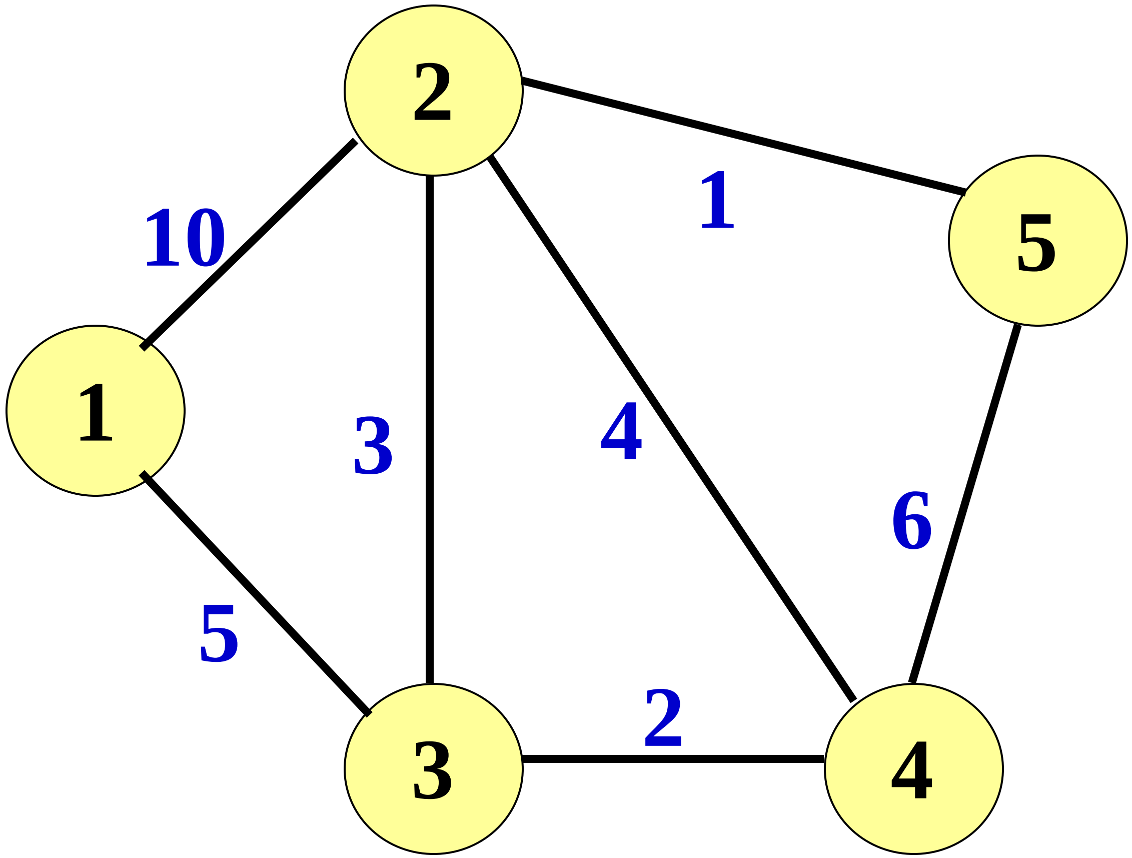


- このうち木となるのは8通りである。



最小木問題

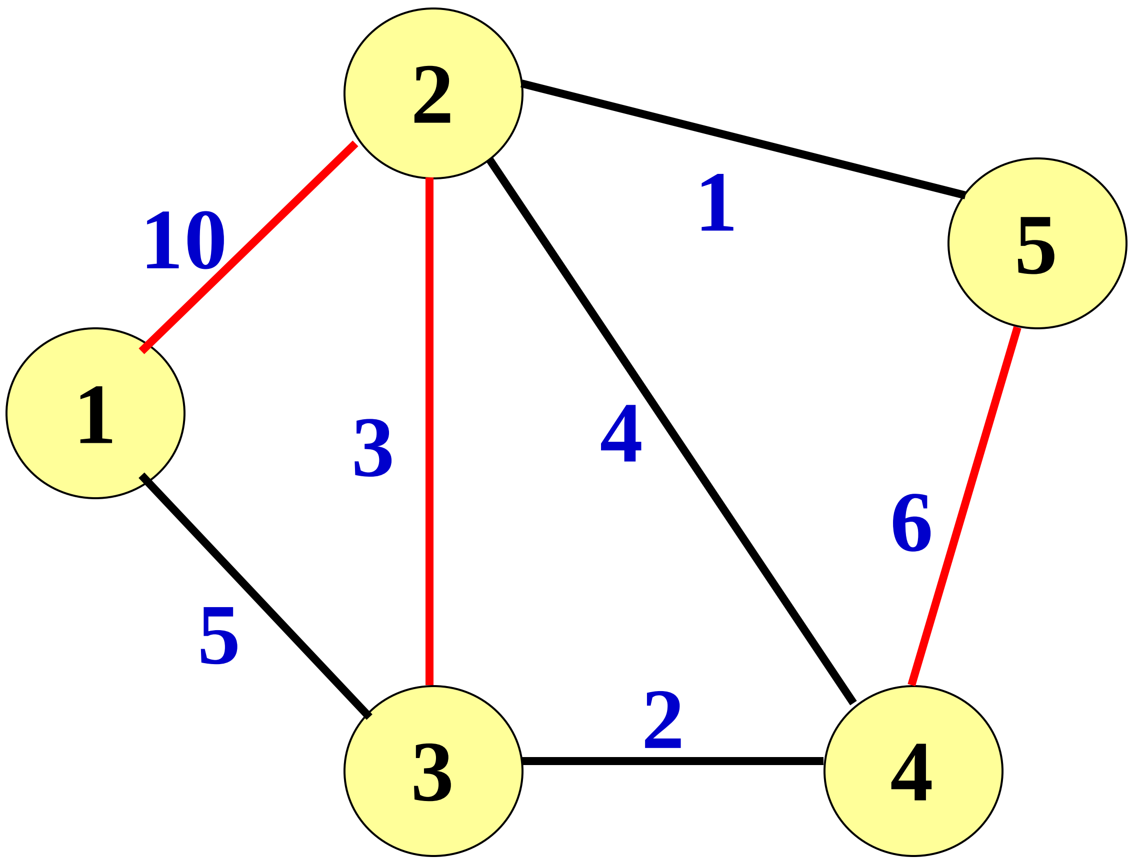
無向グラフ上で、すべての点を張る最小コストの木(閉路を構成しない、連結な枝の集合)を選ぶ問題



各枝の数字は、各点間の距離(費用)

最小木問題

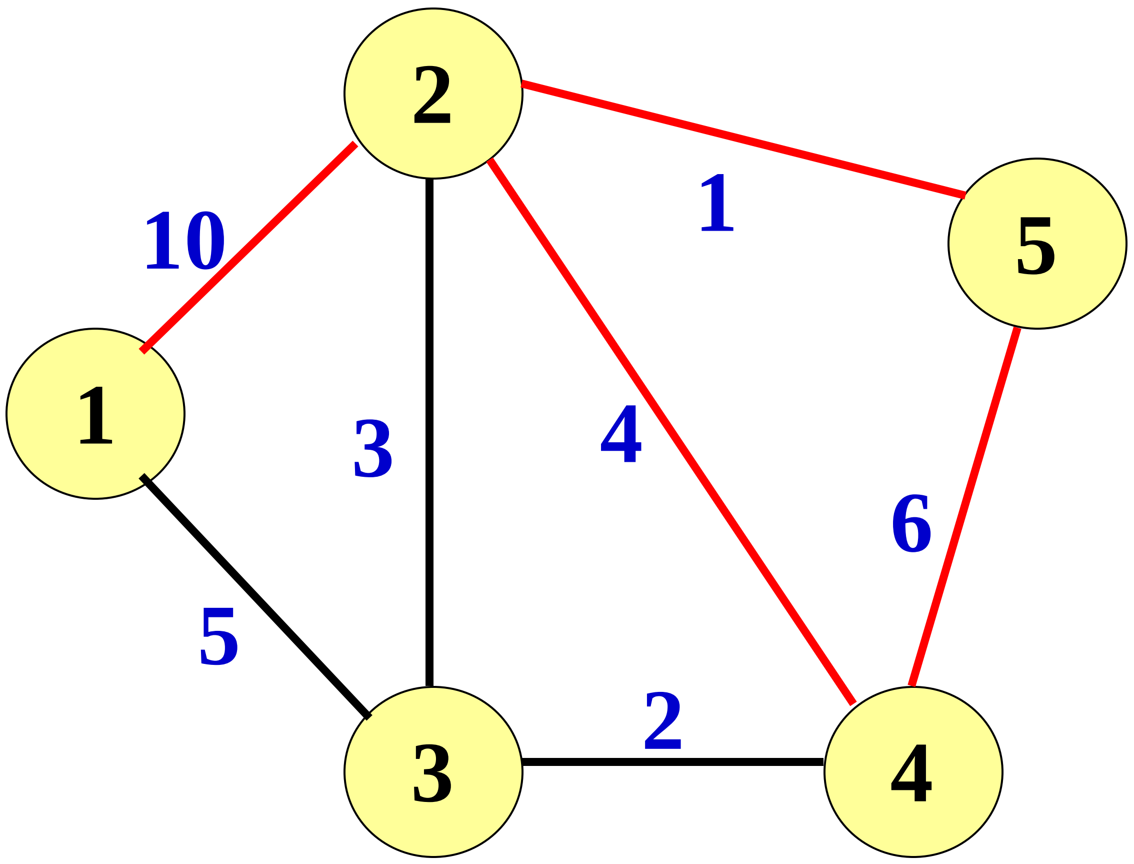
無向グラフ上で、すべての点を張る最小コストの木（閉路を構成しない、連結な枝の集合）を選ぶ問題



木ではない：　連結していない

最小木問題

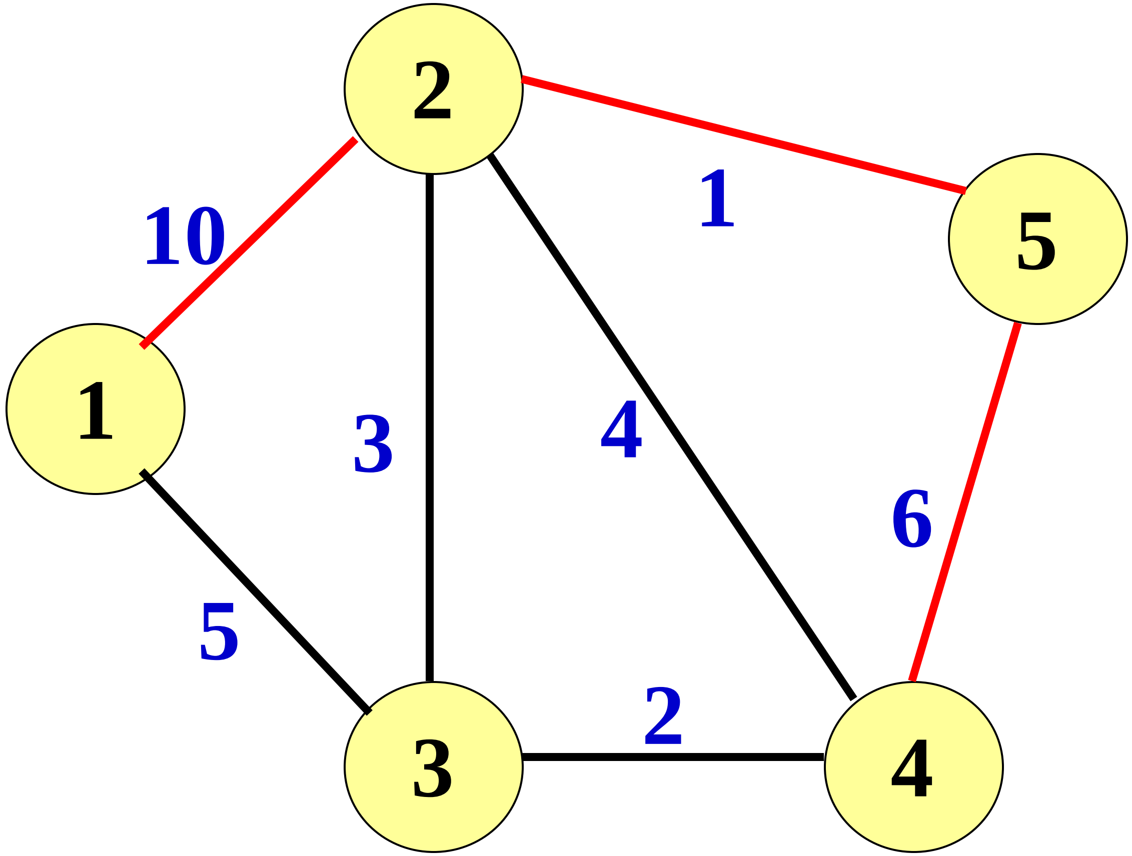
無向グラフ上で、すべての点を張る最小コストの木（閉路を構成しない、連結な枝の集合）を選ぶ問題



木ではない：閉路が存在する

最小木問題

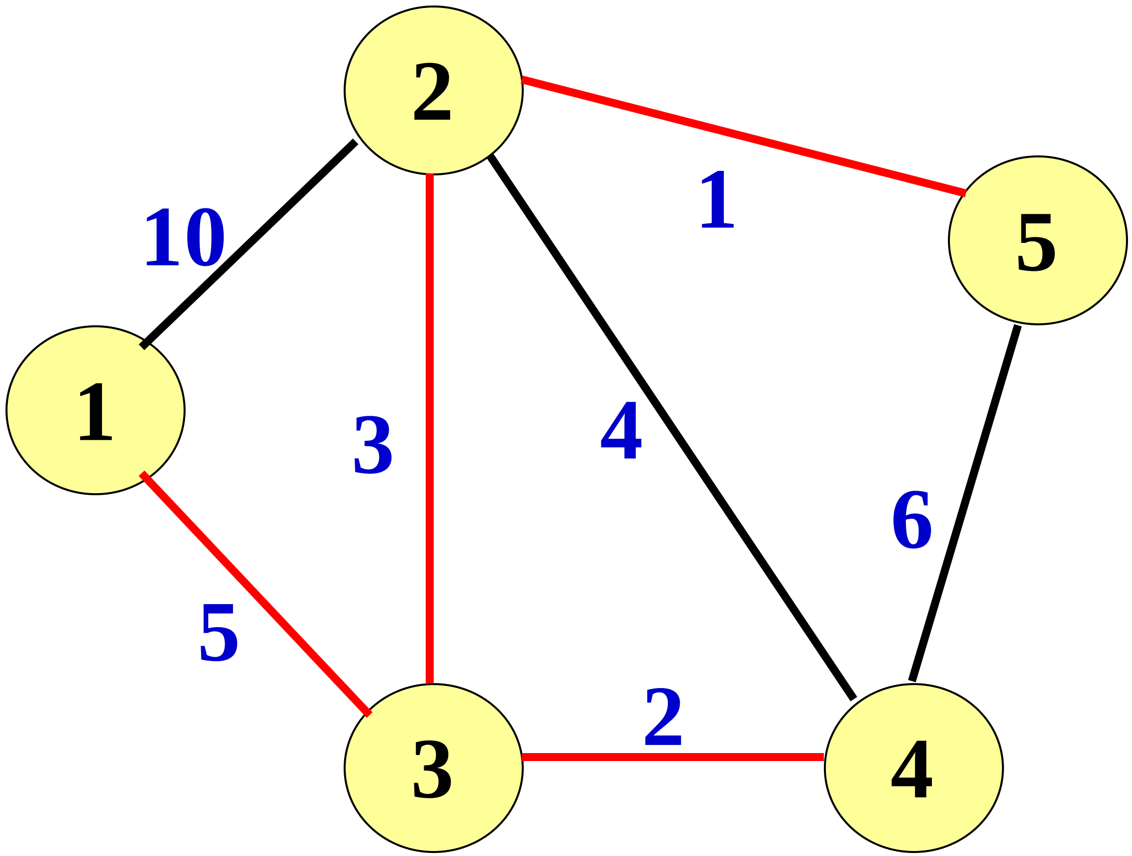
無向グラフ上で、すべての点を張る最小コストの木（閉路を構成しない、連結な枝の集合）を選ぶ問題



すべての点を張らない木

最小木問題

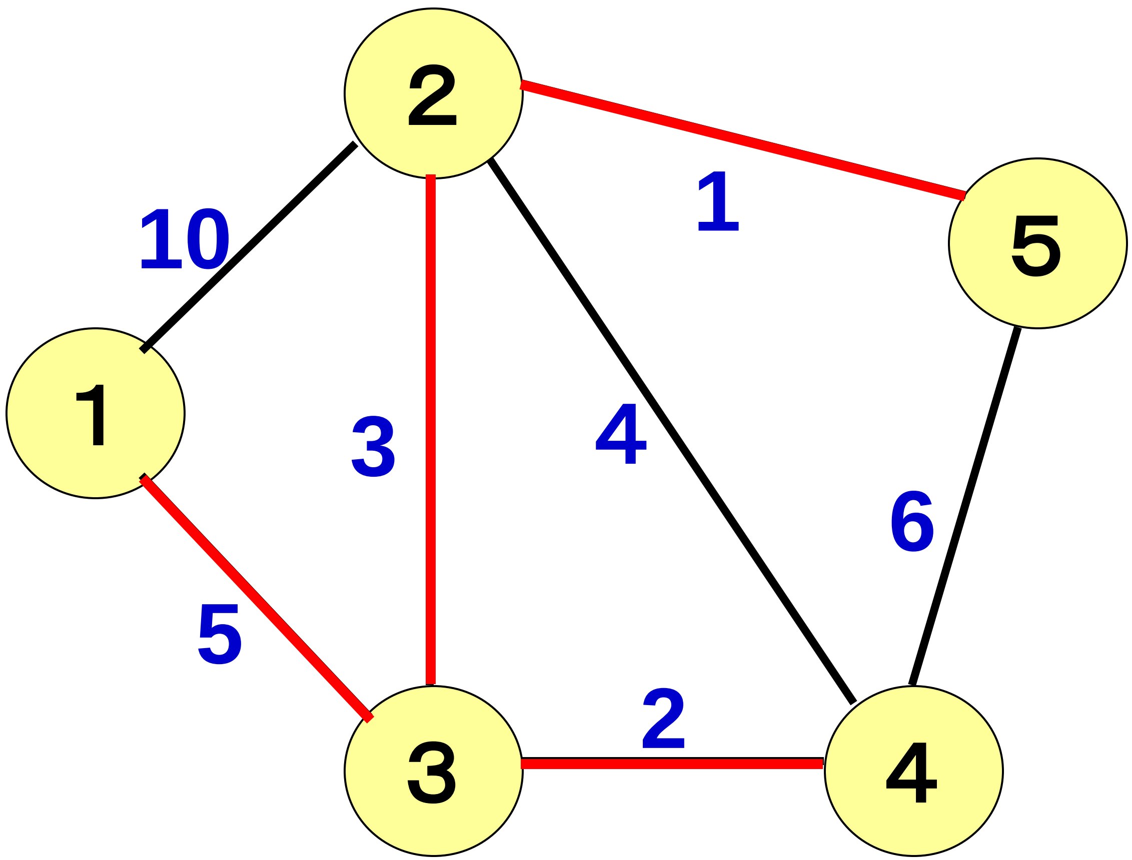
無向グラフ上で、すべての点を張る最小コストの木（閉路を構成しない、連結な枝の集合）を選ぶ問題



最小木：各枝の数字は、各点間の距離（費用）

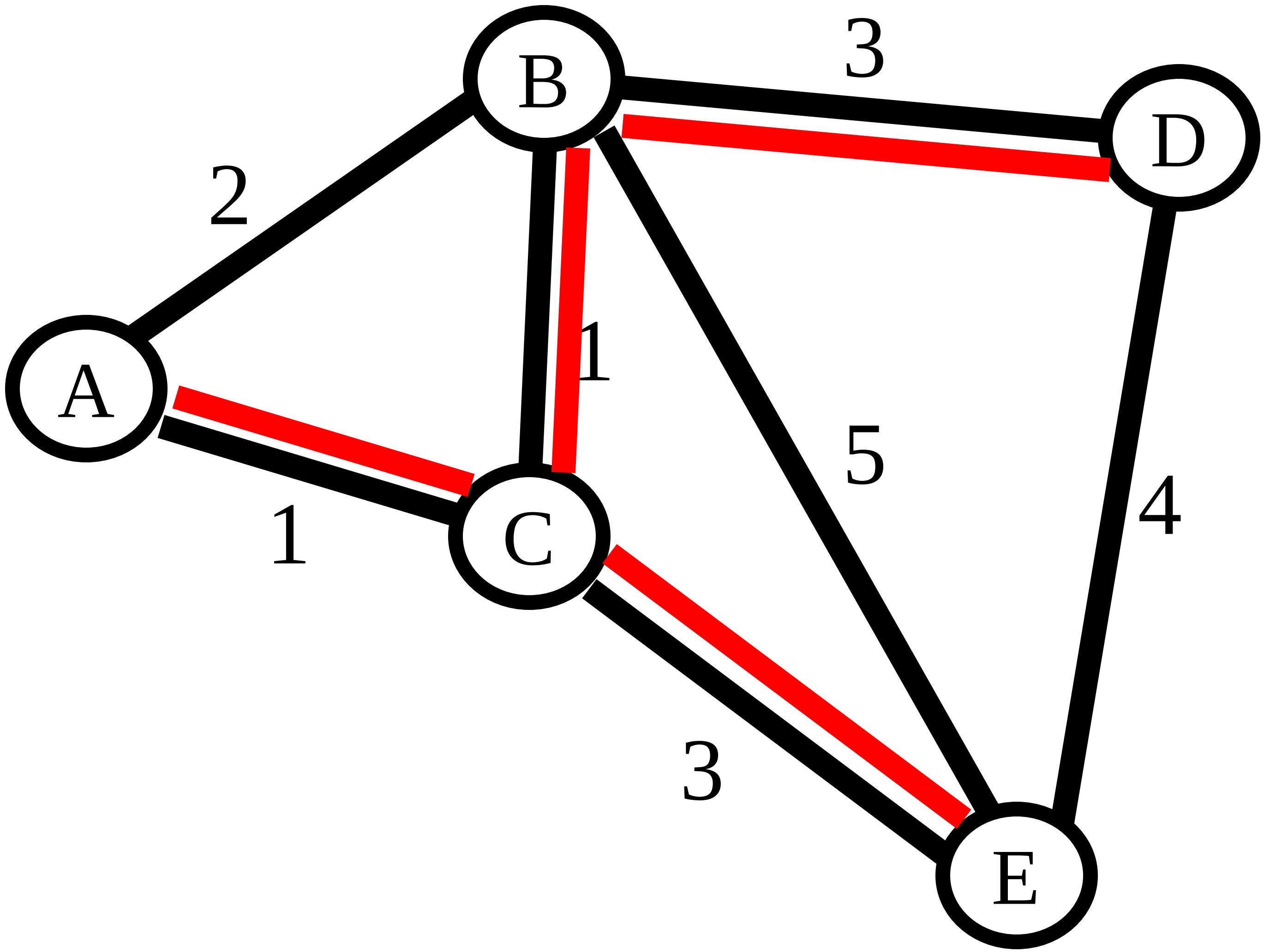
最小木問題

無向グラフ上で、すべての点を張る最小コストの木（閉路を構成しない、連結な枝の集合）を選ぶ問題

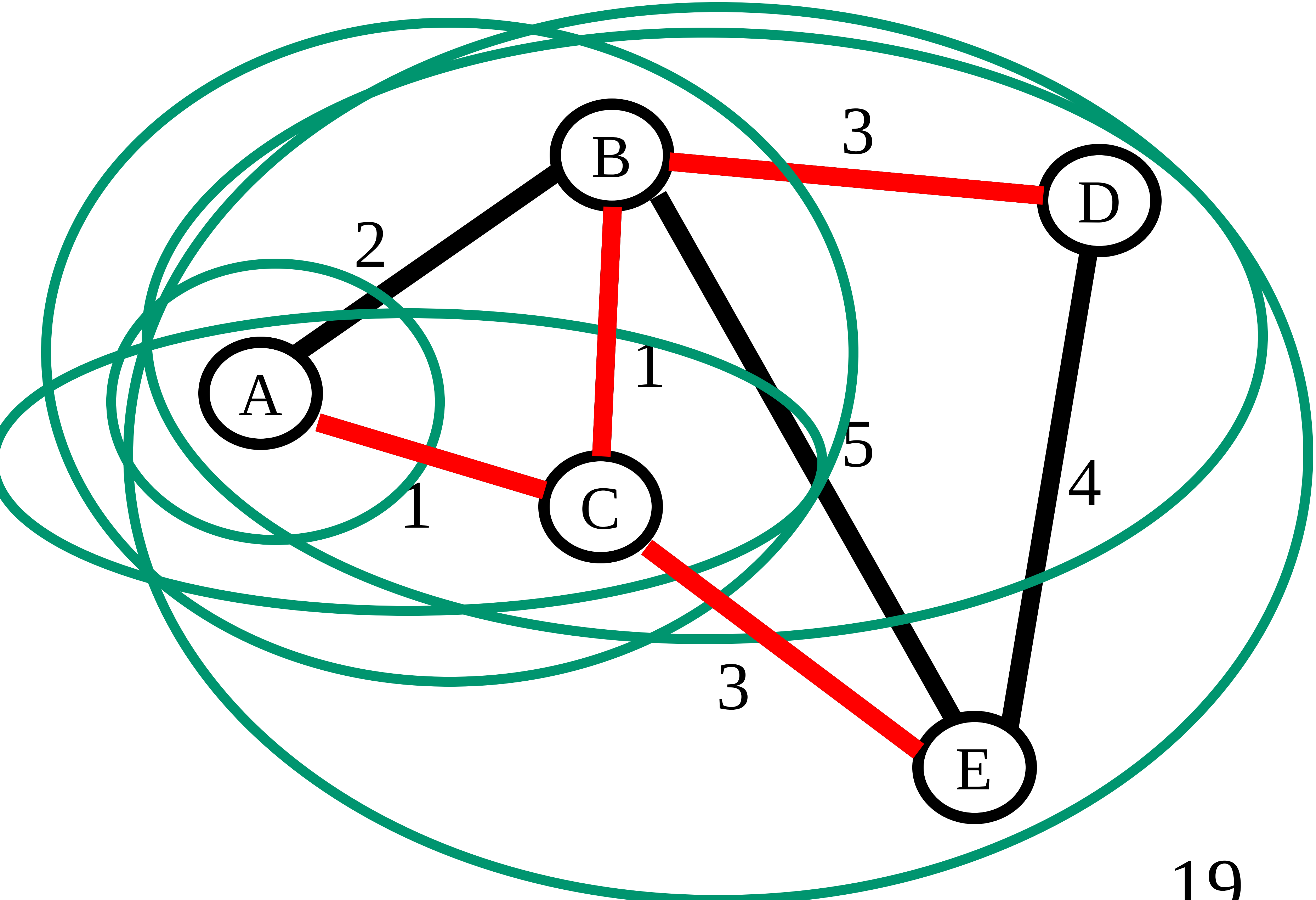


各枝の数字は、各点間の距離（費用）

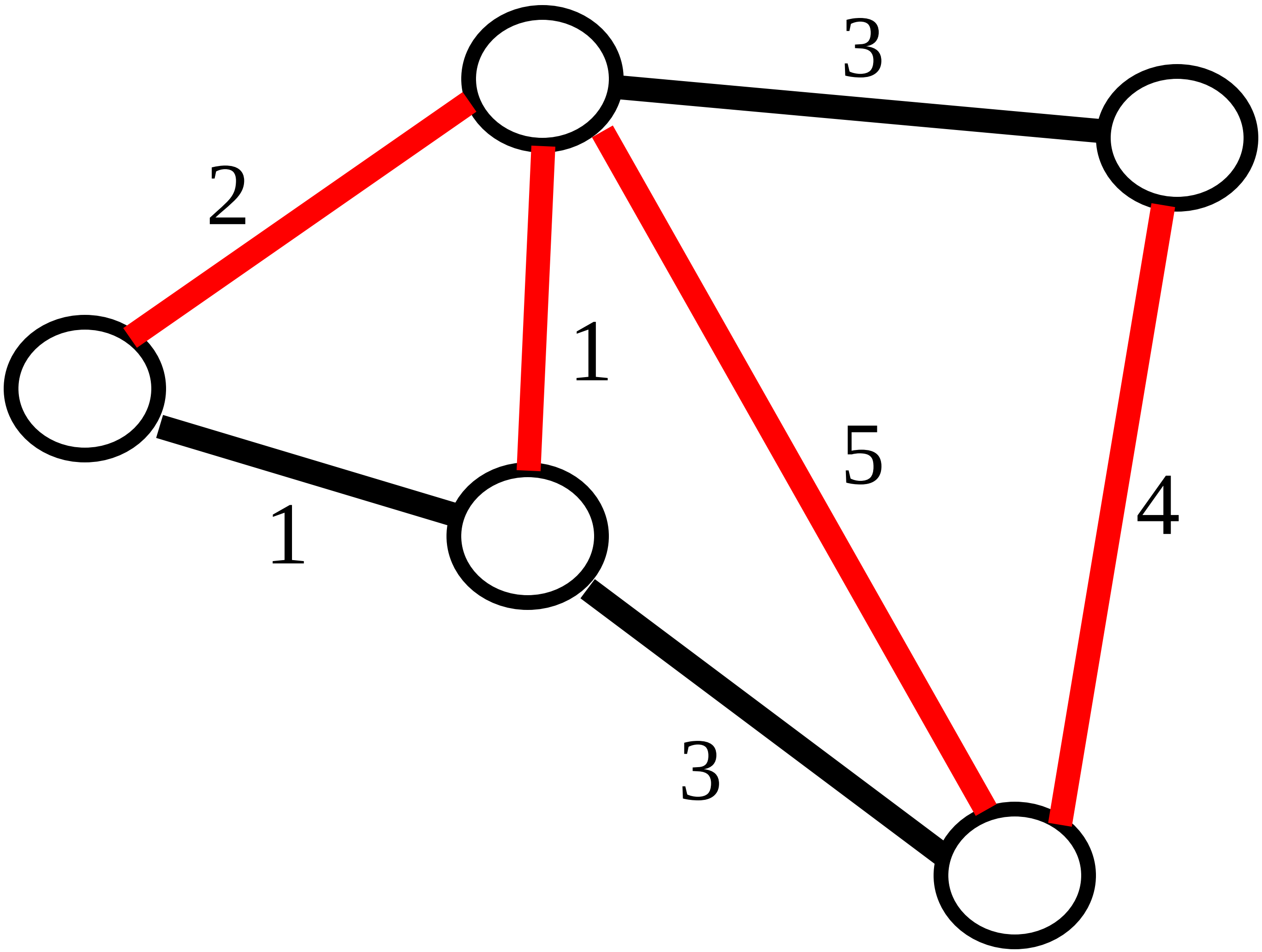
貪欲アルゴリズム（Kruskal法）



貪欲アルゴリズム（Prim法）

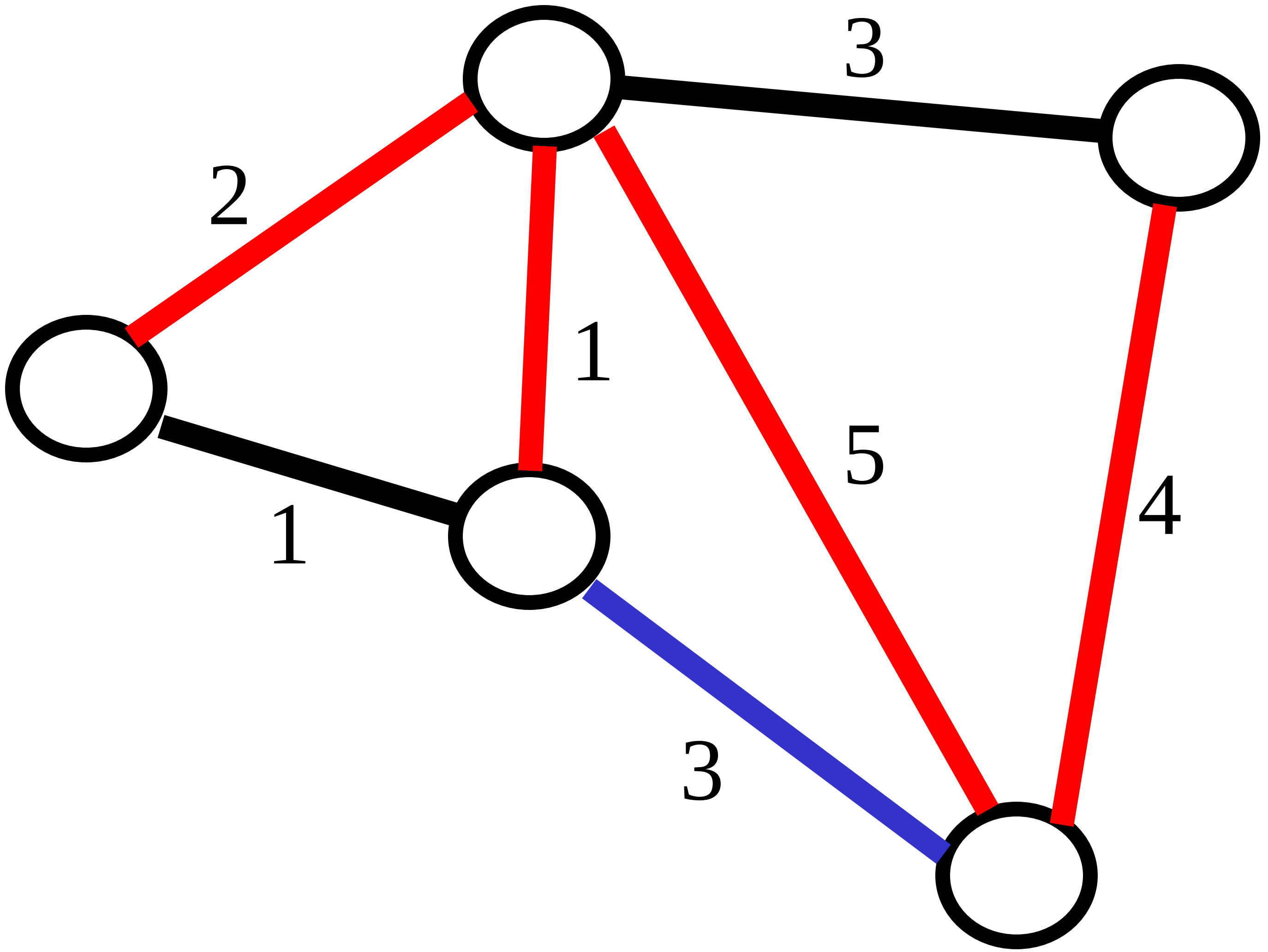


改善法（初期解生成）

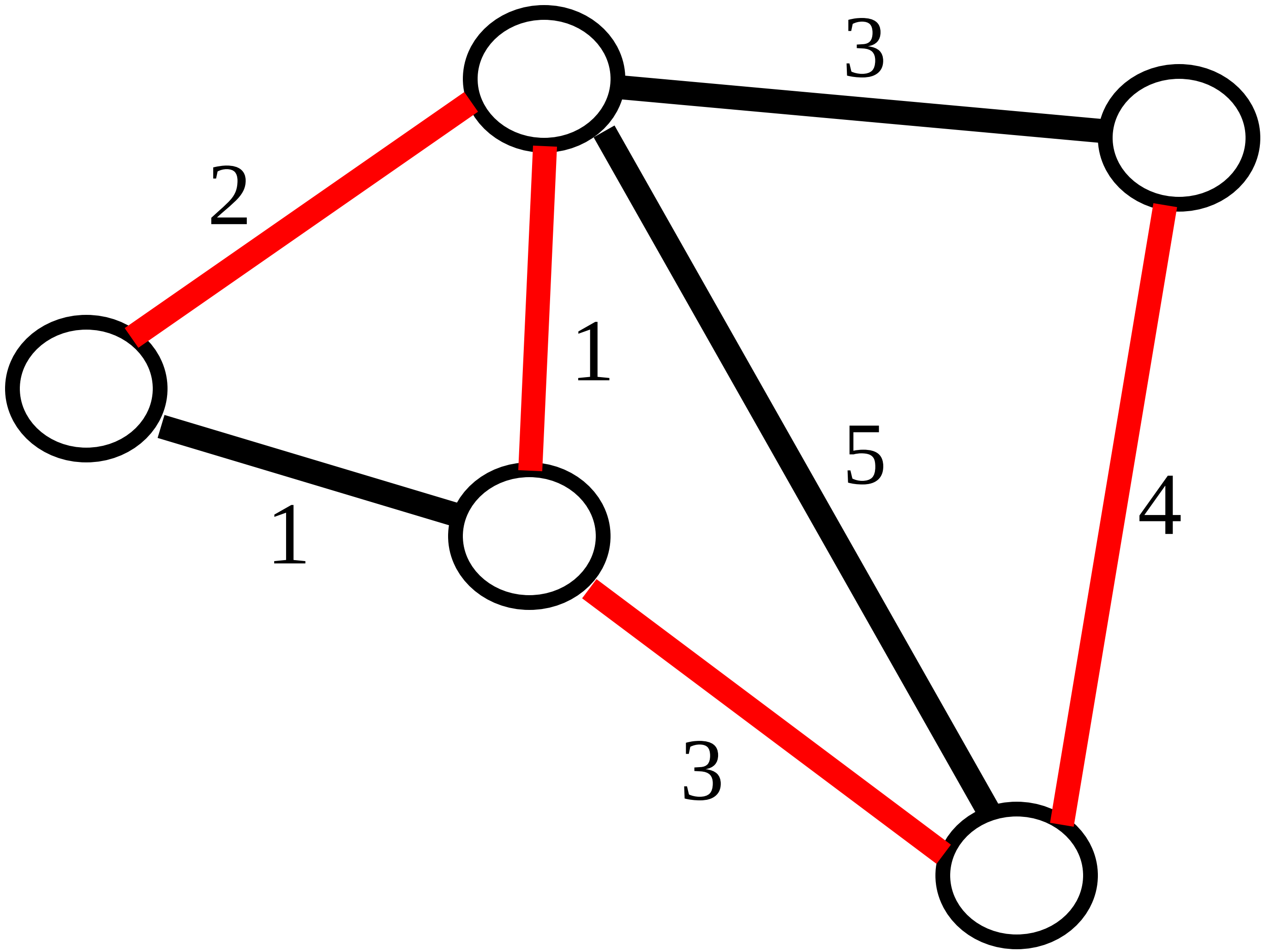


目的関数値　12

改善法

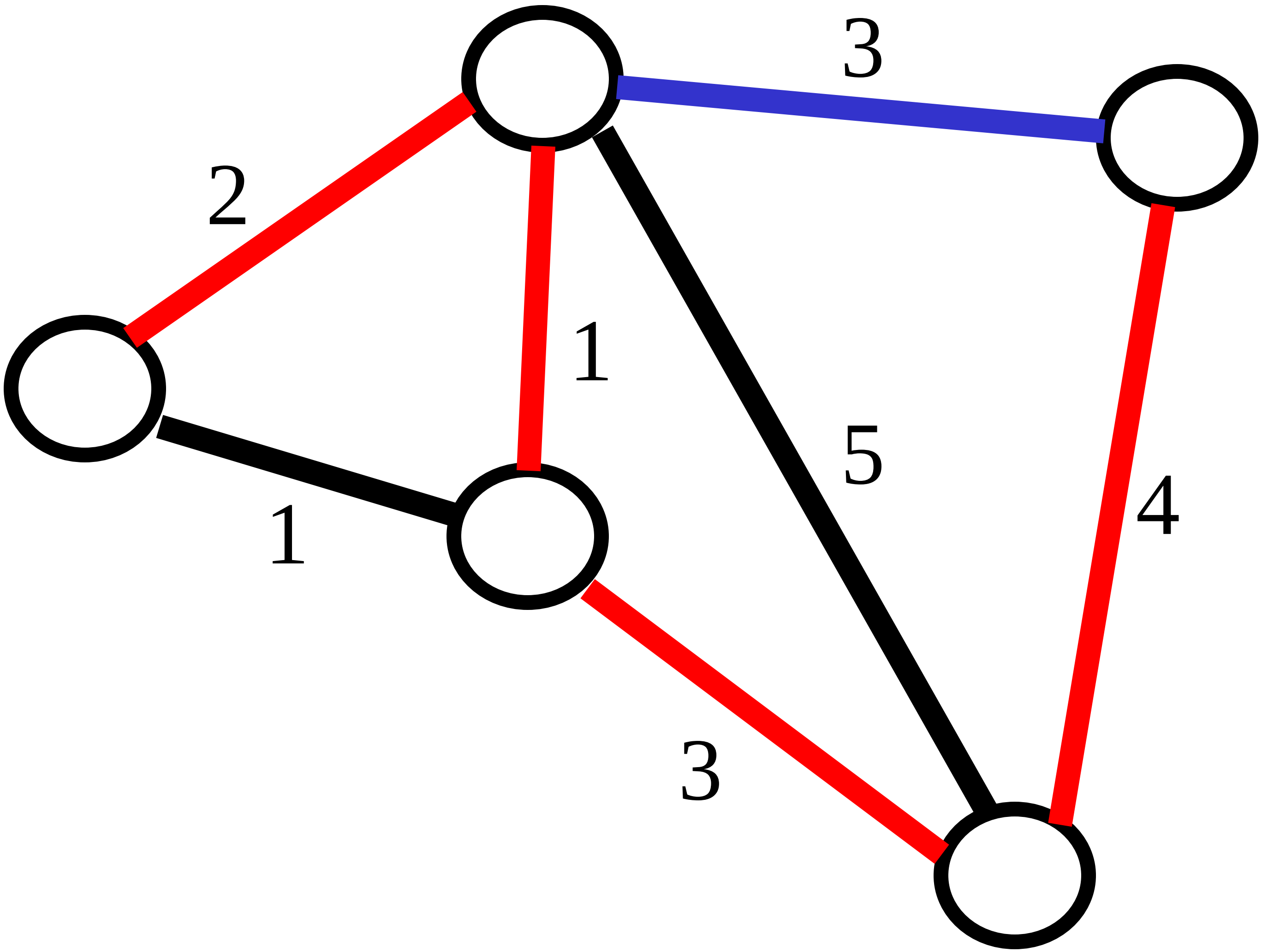


改善法

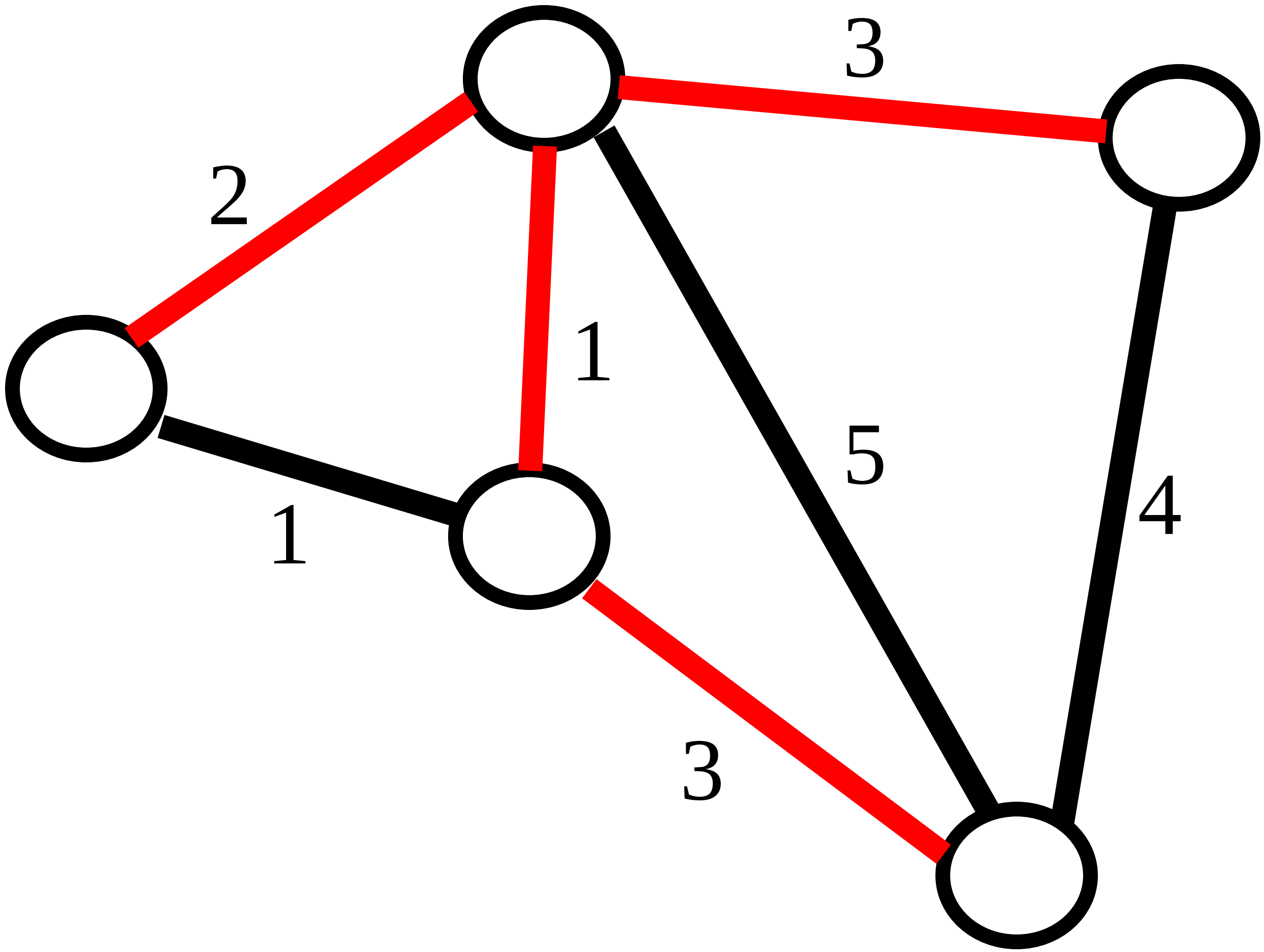


目的関数値　10(12+3-5)

改善法

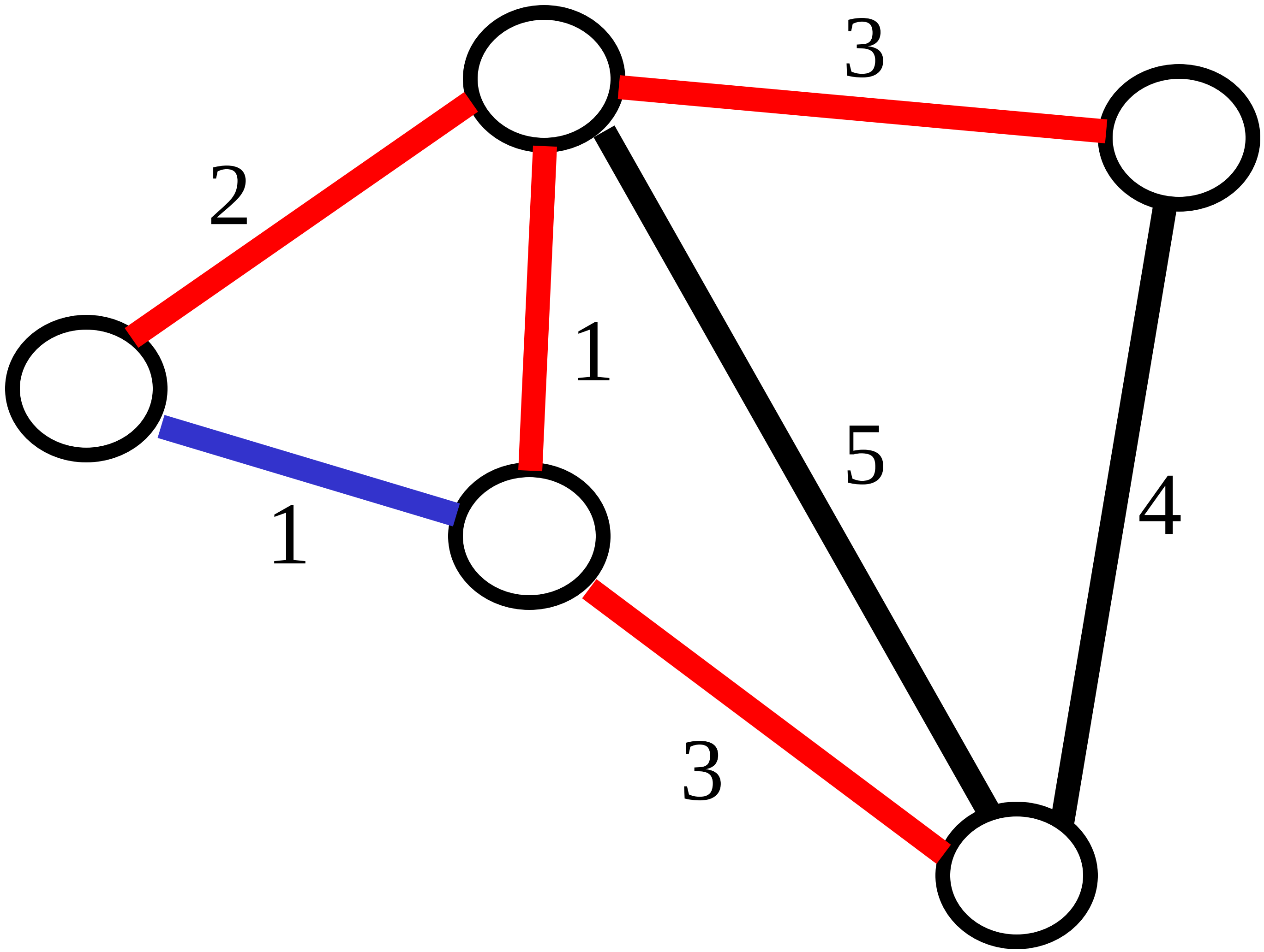


改善法

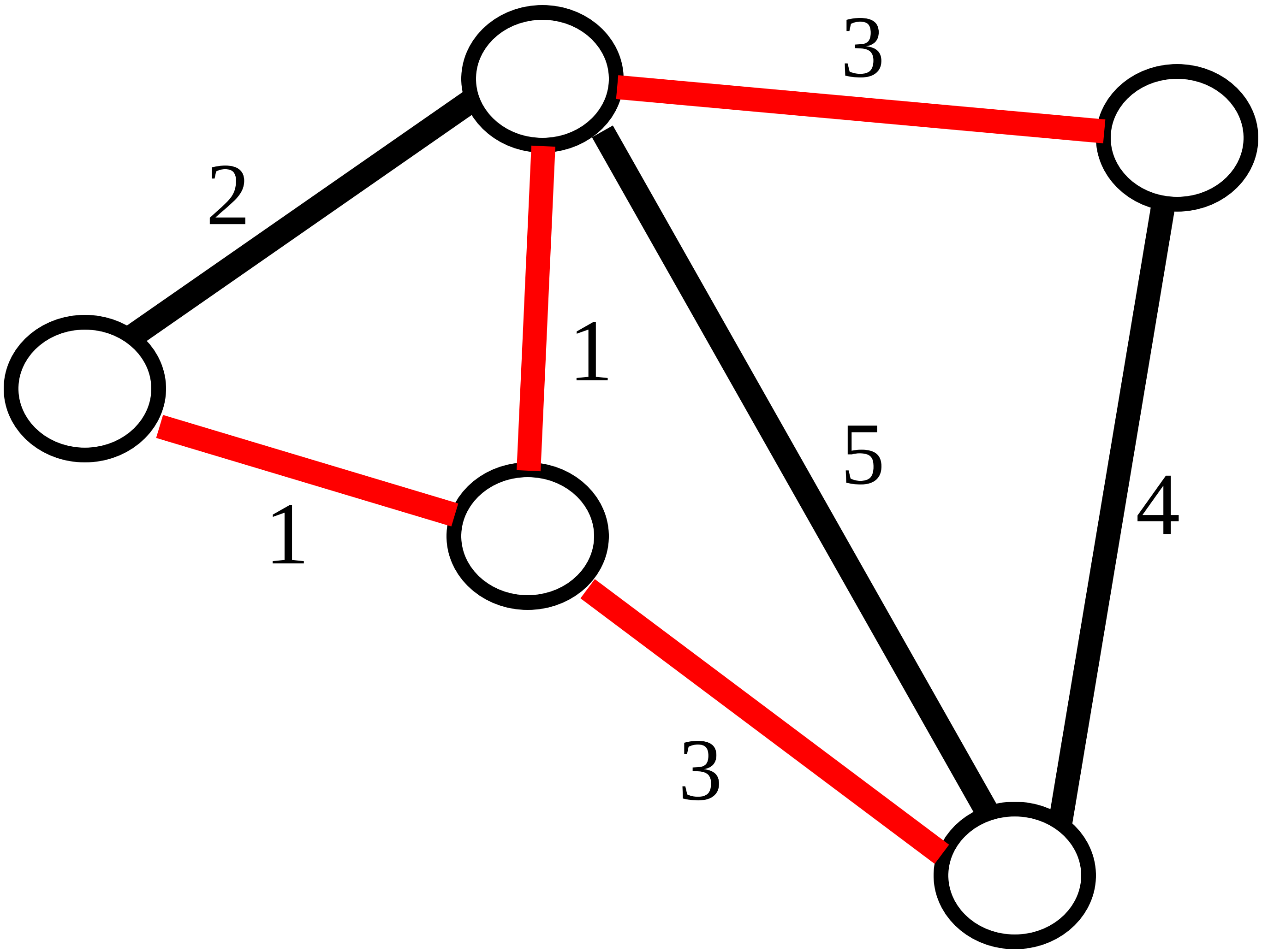


目的関数値　9(10+3-4)

改善法



改善法



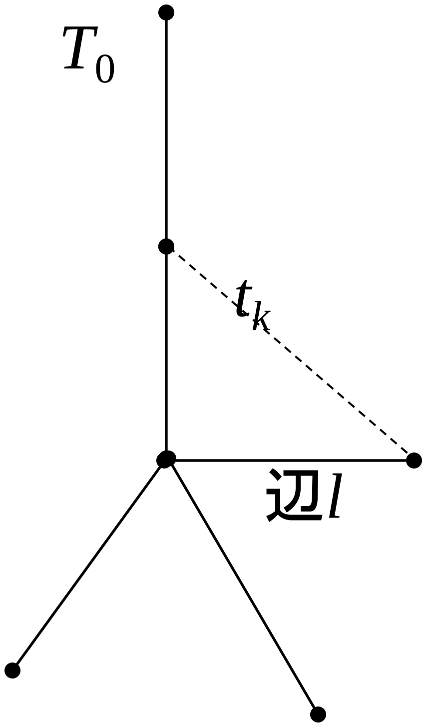
目的関数値　8(9+1-2)

構築法と改善法

- 構築法
  解の存在しない状態から、順次、解を作成していく解法
  例：貪欲解法（最小木問題に対するKruscal法やPrim法など）
- 改善法
  与えられた実行可能解から出発して、順次、よりよい解を見つけていく解法
  例：局所探索法（最小木問題に対する改善法、線形計画問題に対する単体法など）

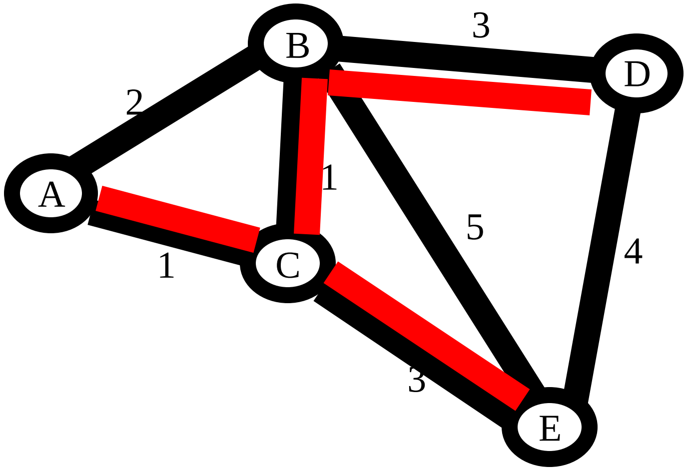
貪欲解法の妥当性（略証）

- 最適解が求められることを示す
- Kruskalの方法で得られた木をT、第k反復でTに加えられる辺をt_kとする
- 任意の最小木T_0を考え、辺$t_k \notin T_0$となる最早の反復kを考えると、$T_0 \cup \{t_k\}$は1つの閉路を必ず含み、その閉路中のt_kとは異なる辺をlとする
- 新たな木$(T_0 - \{l\}) \cup \{t_k\}$を考えると、辺$t_k$の長さ≦辺$l$の長さとなる
- もし、辺t_kの長さ＜辺lの長さとなる場合は、新たな木$(T_0 - \{l\}) \cup \{t_k\}$の長さは$T_0$よりも小さくなるため、最小性に反する
- よって「辺t_kの長さ=辺lの長さ」となるため、Tの重みとT_0の重みは変わらない。（この手続を繰り返す）



貪欲解法における閉路の検出

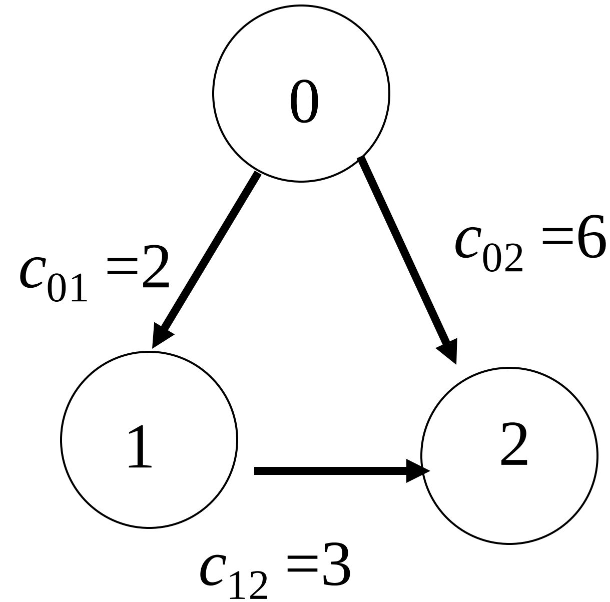
- 重みの小さい順に辺を加えるとき、閉路が生成されてはいけない。
- 図を見ると、閉路が存在するかどうかすぐわかる
- 実際の大規模データ、実ネットワークでは点の位置は座標のみ、図示されていないことも多く、データ数も膨大
- 閉路検出法



- 実装（プログラミング）は効率的に行うことが可能

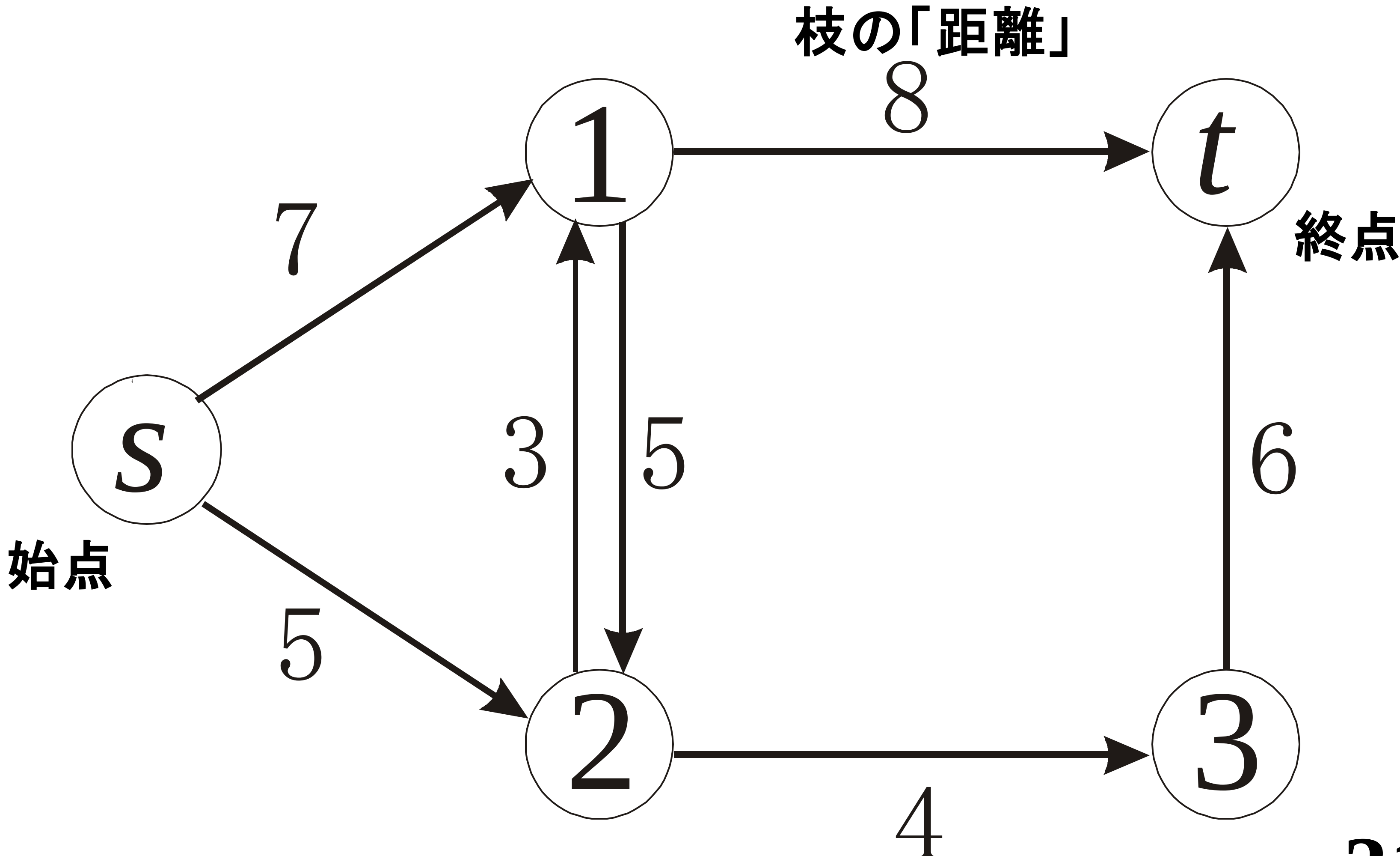
最短路問題:

- <u>ネットワークに関する定義</u>
- 有向グラフ G=(V,A)を考える（V: 点集合、A: 辺集合）
- 各辺$(i, j) \in A$に対して辺の長さc_{ij} が与えられている
- 例: 点集合V={0,1,2}辺集合A={(0,1),(1,2),(0,2)}
- 点0からすべての点への最短路を求めたい
- 次のπ_jを点0から点jへの最短距離としよう

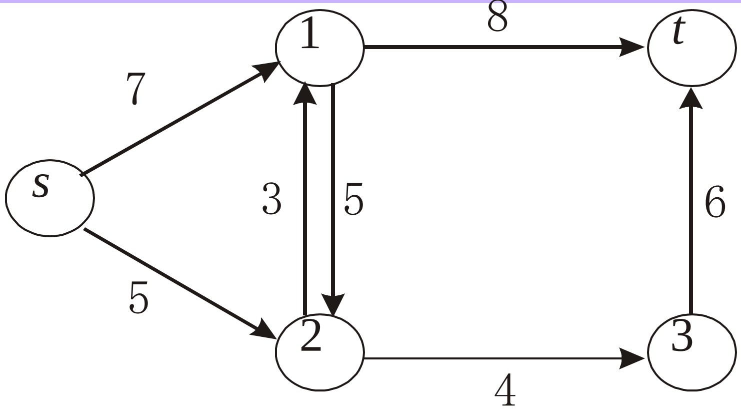


(i) 任意の辺(i,j)について、 $\pi_j \leqq \pi_i + c_{ij}$
(点0から点 j までの最短路は点 j の
直前に点 i を通るとは限らない）
(ii)任意の0でない点 j ($\neq$0)に対して、
$\pi_j = \pi_i + c_{ij}$ となる点 i が存在する。
（点0から点 j までの最短路は点 j の
直前に点 i を通る）

最短路問題
（始点sから終点tへの最短路を求める）



最短路問題の
線形計画問題による定式化



最小化 $7x_{s1}+5x_{s2}+5x_{12}+8x_{1t}+3x_{21}+4x_{23}+6x_{3t}$

条件

$$-x_{s1}-x_{s2}=-1$$
$$x_{s1}-x_{12}-x_{1t}+x_{21}=0$$
$$x_{s2}+x_{12}-x_{21}-x_{23}=0$$
$$x_{23}-x_{3t}=0$$
$$x_{1t}+x_{3t}=1$$

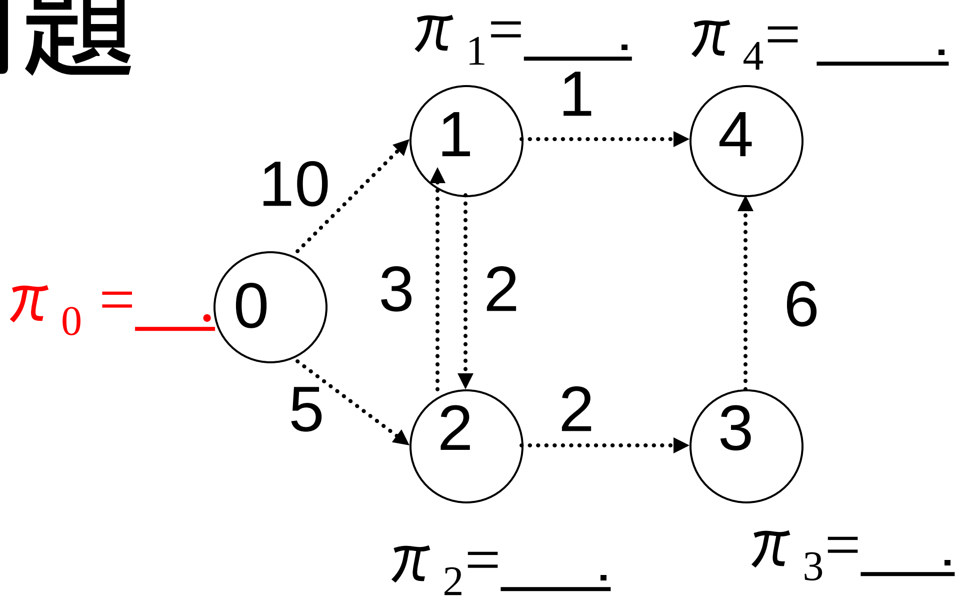
$x_{s1}, x_{s2}, x_{12}, x_{1t}, x_{21}, x_{23}, x_{3t} \geq 0$ 必ず0-1条件を満たす解が得られる

Dijkstra（ダイクストラ）法

- ステップ1： $\pi_0=0, \pi_j=\infty$ $(j \in V-\{0\})$,
  $M=\{1,2,\ldots,n\}$, 探索開始点 $i=0$
- ステップ2：$M \neq \varphi$（空集合）でない場合(i)(ii)(iii)を繰り返す
  (i) $j \in M$ に到達する直前点を可能なら仮に i とする
  $j \in M$ に対し $\pi_j > \pi_i + c_{ij}$ ならば $\pi_j = \pi_i + c_{ij}$, $p_j = i$
  (ii) $j \in M$ への暫定的最短距離から最小の
  $\min_{j \in M} \pi_j = \pi_k$ となる k を求める
  (iii) k を M から除く、$i=k$ とする(k からの探索を行う)

•M：最短路未確定な点の集合（M から削除されると最短路確定）
•p_j：点 j への最短路の直前点（終了時）
•点 i：探索を行う点(i は最短路決定、i を直前点とする探索)

例題



- 辺上の数字は辺の長さ

ステップ1：

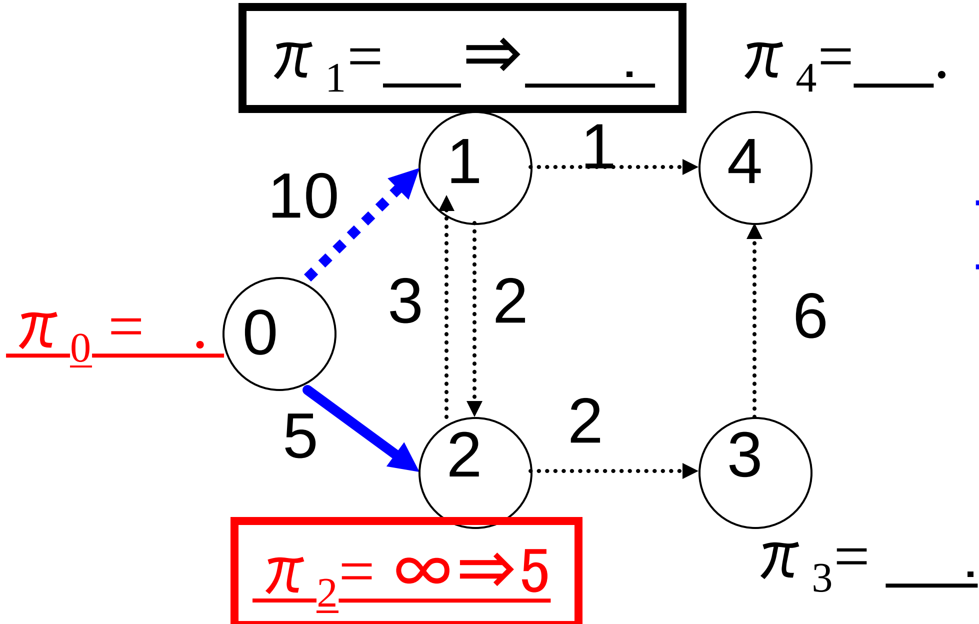
π_0=___, π_i=___, ($i \in M$),

M={____________}

点___のみ最短路確定

i=__ よりスタート

•M：最短路未確定
•Mから出ると最短路確定



青い実線:最短路が決まった
青い点線:最短路候補が決まった

ステップ2(1回目): i=___, M={_________}

(i) M={_________}に到達する直前点を可能なら仮に i=0とする

　j=1については π_1　　　　　　　　なので π_1=___________, p_1=___.

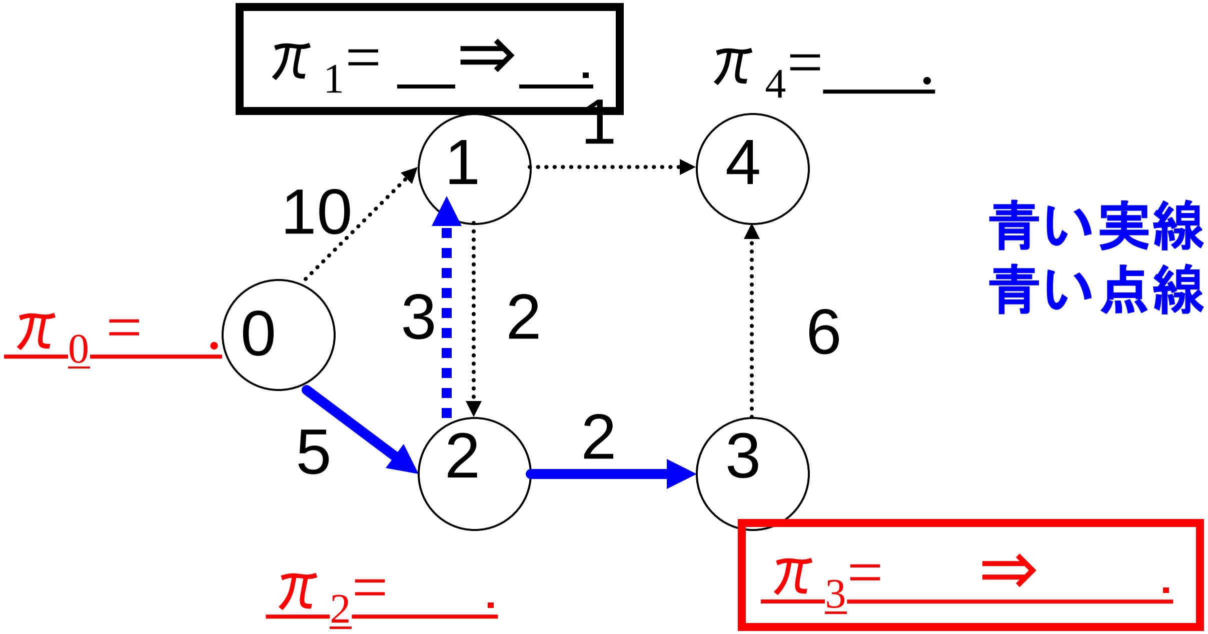
　j=2については π_2　　　　　　　　なので π_2=___________, p_2=___.

　j=3,4については、直接i=　　と接続していないのでそのまま

(ii) $\min_{j\in M}\ \pi_j = \min$ {_____________} = ____.

点k=__の最短路確定: 1に到達する直前点が p_k =___となる

(iii) k=___ をMから削除し、i=___より再びステップ2を行う



青い実線:最短路が決まった
青い点線:最短路候補が決まった

ステップ2(2回目):i=__, M={__________}

(i) M={_____}に到達する直前点を可能なら仮に i=__とする

j=1についてはπ_1__________なのでπ_1= __________, p_1=___.

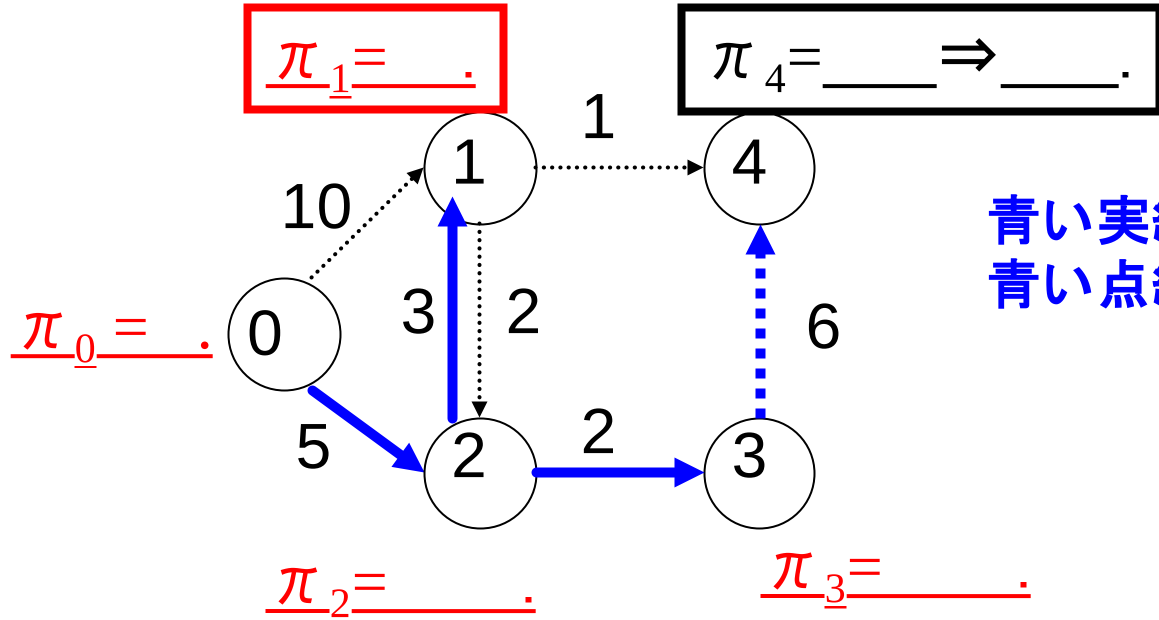
j=3についてはπ_3__________なのでπ_3= __________, p_3=___.

j=4については、直接i=__と接続していないのでそのまま

(ii) $\min_{j\in M}\ \pi_j = \min\ \{$__________$\}$ = ____.

点k=__の最短路確定: ___に到達する直前点が p_k =____ となる

(iii) k=__をMから削除し、i=__より再びステップ2を行う



青い実線:最短路が決まった
青い点線:最短路候補が決まった

ステップ2(3回目): i=____ , M={_______}

(i) M={_____}に到達する直前点を可能なら仮に i= ___とする

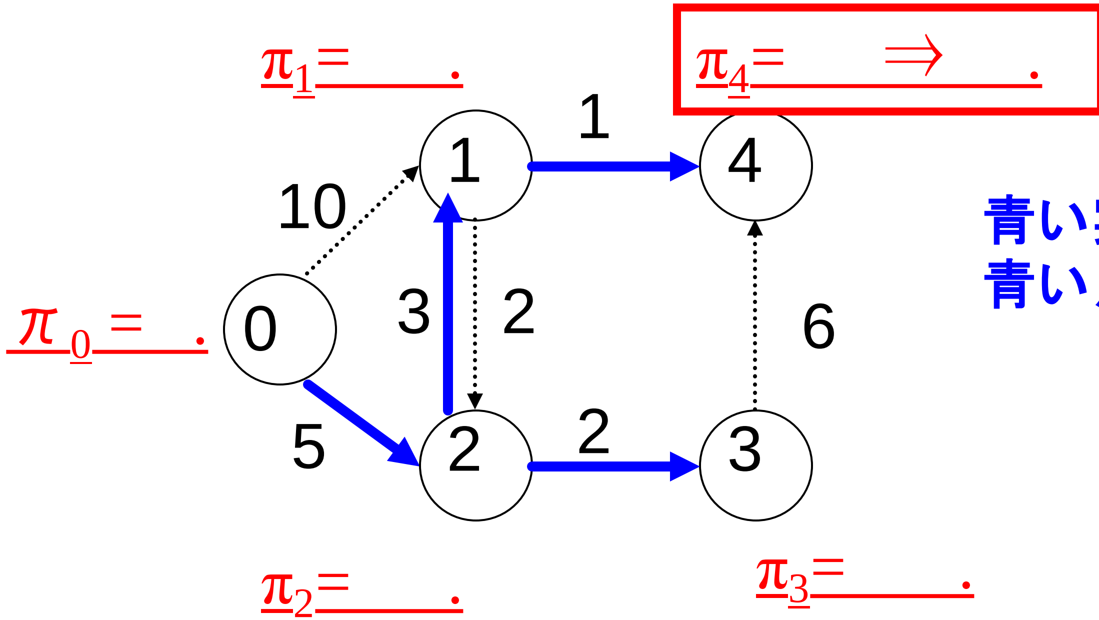
j=1については、直接i=___と接続していないのでそのまま

j=4についてはπ_4 ______________なのでπ_4=_________, p_4=___.

(ii) $\min_{j\in M} \pi_j$ =min {________} =____.

点k=__の最短路確定: __に到達する直前点が p_k =___となる

(iii) k=__をMから削除し、i=__より再びステップ2を行う



青い実線：最短路が決まった
青い点線：最短路候補が決まった

ステップ2（4回目）: i=___, M={___}

(i) M={____}に到達する直前点を可能なら仮に i=___とする

j=___についてはπ_4____________なのでπ_4=__________, p_4=___.

(ii) $\min_{j\in M}\pi_j$ =min {___} =_____.

点k=___の最短路確定: ____に到達する直前点がp_k =___となる

(iii) k=___ をMから削除するとMは空集合になるため、最短路が得られた

双対問題（標準型）

（P）　min　$z = c^T x$

s.t.　$Ax = b$

$x \geqq 0$

（D）　max　$w = y^T b$　$(w = b^T y)$

s.t.　$y^T A \leqq c^T$　$(A^T y \leqq c)$

y の符号条件なし

双対定理：　min　$z = c^T x \geqq \max w = y^T b$

相補スラック定理：　x, yが最適解⇔

$$y^T(Ax - b) = 0,\ (y^T A - c^T)x = 0$$

最短路問題の双対問題

$$\max \ \pi_t - \pi_s$$

$$\text{subject to } \pi_j - \pi_i \leq c_{ij}, \forall (i, j) \in A$$

- 任意のaについてπ_iを$\pi_i + a$ と置き換えても実行可能⇒ 始点で$\pi_s=0$と定める
- π_iとは？⇒始点sから点iへの最短距離の下界
- π_iをsからiへの最短距離と定めると双対問題の実行可能解なので、双対問題を解いてもsからtへの最短距離が求められることがわかる

最短路問題：双対問題

max $\pi_4 - \pi_0$

s.t. $\pi_1 - \pi_0 \leq 10$

$\pi_2 - \pi_0 \leq 5$

$\pi_2 - \pi_1 \leq 2$

$\pi_4 - \pi_1 \leq 1$

$\pi_1 - \pi_2 \leq 3$

$\pi_3 - \pi_2 \leq 2$

$\pi_4 - \pi_3 \leq 6$

- $\pi_j - \pi_i \leqq c_{ij}$
- 点j までの最短路の距離

  $\leqq$

  （点iまでの最短路の距離

  ＋(i, j) 間の距離）

最短路問題での相補スラック定理

相補スラック定理： x,yが最適解⇔

$$y^{\mathrm{T}}(Ax-b)=0\text{（明らか）},\ (y^{\mathrm{T}}A-c^{\mathrm{T}})x=0$$

$(\pi 1-\pi 0-10)x01=0 \;\Rightarrow$

$(\pi 2-\pi 0-5)x02=0 \;\Rightarrow$

$(\pi 2-\pi 1-2)x12=0 \;\Rightarrow$

$(\pi 1-\pi 2-3)x21=0 \;\Rightarrow$

$(\pi 4-\pi 1-1)x14=0 \;\Rightarrow$

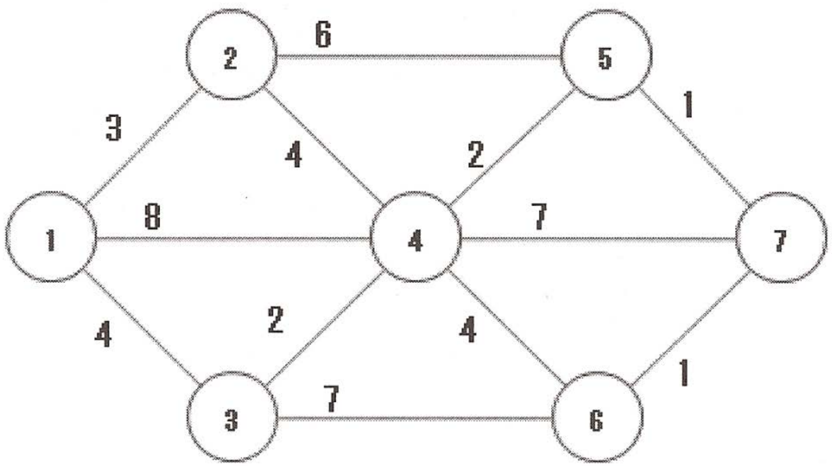
$(\pi 3-\pi 2-2)x23=0 \;\Rightarrow$

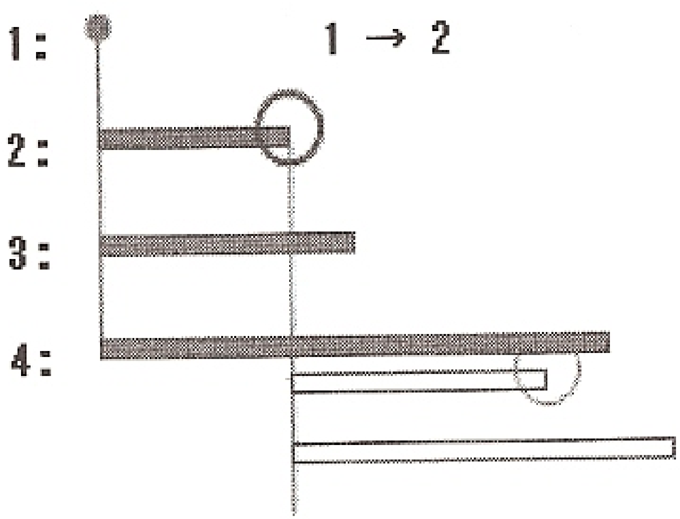
$(\pi 4-\pi 3-6)x34=0 \;\Rightarrow$

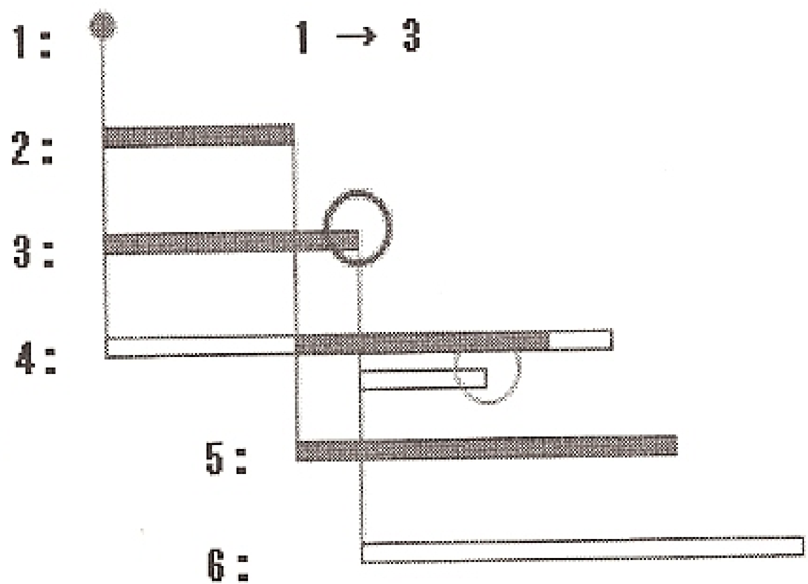
最短路に含まれる辺(i,j)では　　　　　となり、

このとき　　　　　　　　となる。

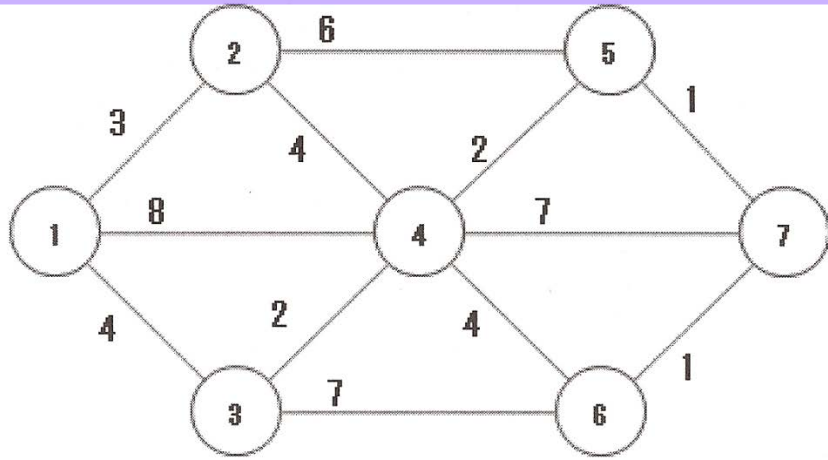
ガントチャートもどきの図による解法

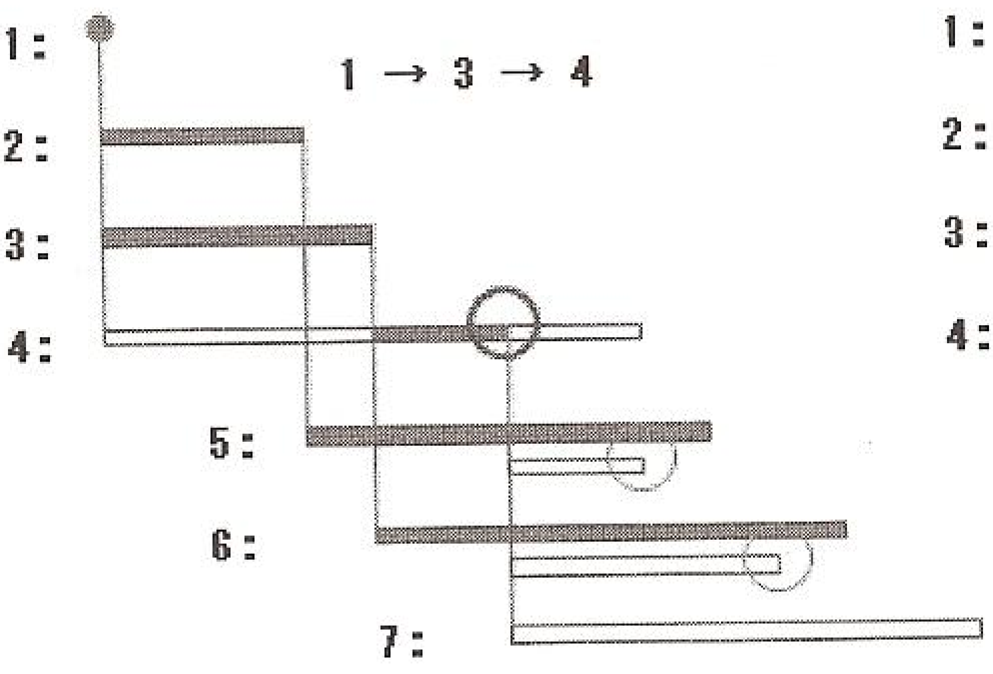


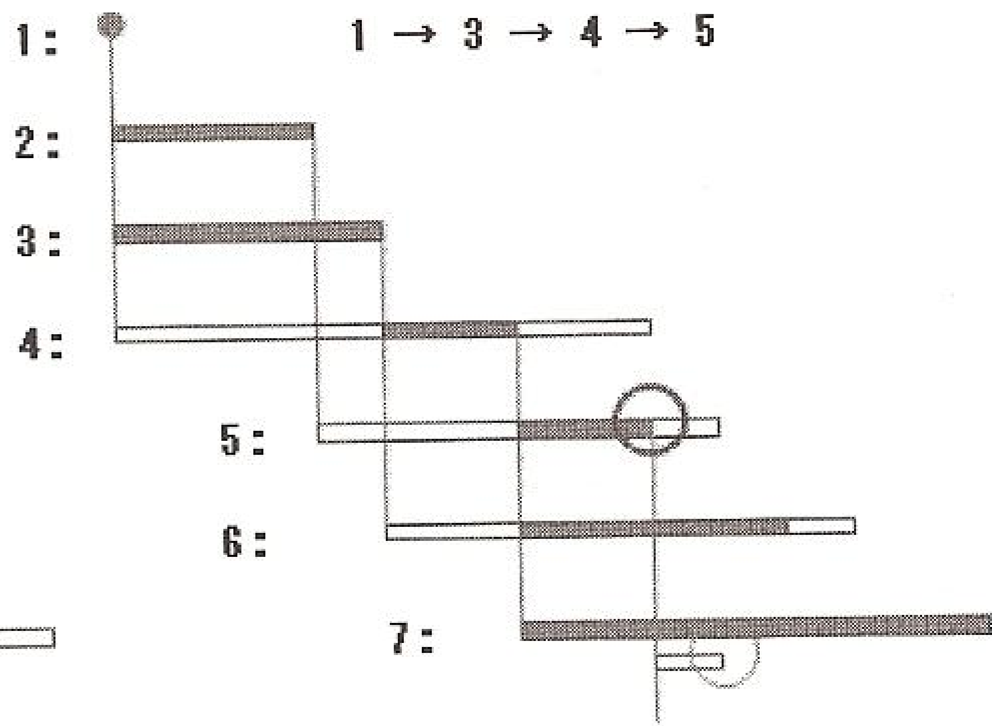




ガントチャートもどきの図による解法



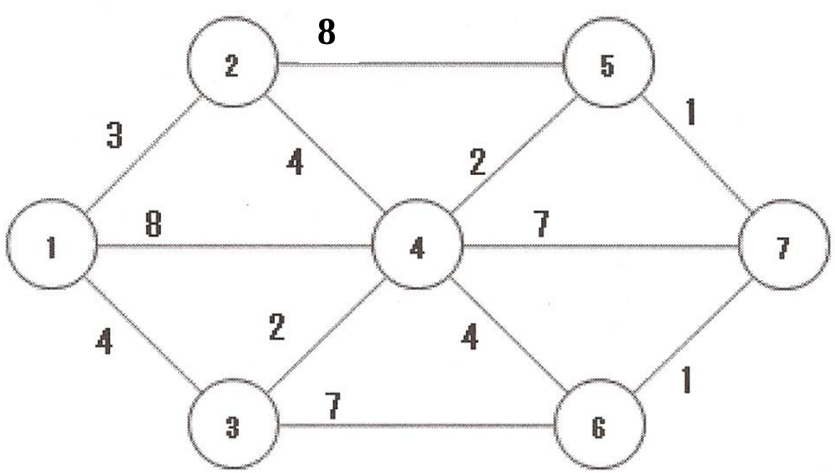




任意の2点間の最短路$d_{ij} \leftarrow \min\{d_{ij},\ d_{ik}+d_{kj}\}$

	1	2	3	4	5	6	7
1	0	3	4	8	∞	∞	∞
2	3	0	∞	4	8	∞	∞
3	4	∞	0	2	∞	7	∞
4	8	4	2	0	2	4	7
5	∞	8	∞	2	0	∞	1
6	∞	∞	7	4	∞	0	1
7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
k=1	1	0	3	4	8	∞	∞	∞
	2	3	0	∞	4	8	∞	∞
	3	4	∞	0	2	∞	7	∞
	4	8	4	2	0	2	4	7
	5	∞	8	∞	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0



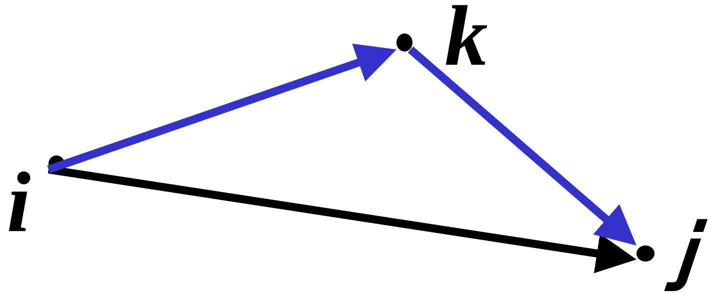
		1	2	3	4	5	6	7
k=1	1	0	3	4	8	∞	∞	∞
	2	3	0	7	4	8	∞	∞
	3	4	7	0	2	∞	7	∞
	4	8	4	2	0	2	4	7
	5	∞	8	∞	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

Warshall-Floyd法 任意の2点間の最短路 三角オペレーション

- 距離行列 $D = [d_{ij}]$
- 点 j に関する三角オペレーション

$$d_{ij} \leftarrow \min\{d_{ij},\ d_{ik} + d_{kj}\}$$

$$\text{for all } i, j = 1, \ldots, n;\quad i, j \neq k$$



Warshall-Floyd法のアルゴリズム 任意の2点間の最短路

- 元々の距離行列 $C=[c_{ij}]$
- 初期化

  for all $i \neq j$ do $d_{ij} \leftarrow c_{ij}$

  for $i = 1,\ldots,n$ do $d_{ii} \leftarrow 0$
- 基本部分

  for $k = 1,\ldots,n$ do

  for $i = 1,\ldots,n$; $i \neq k$ do

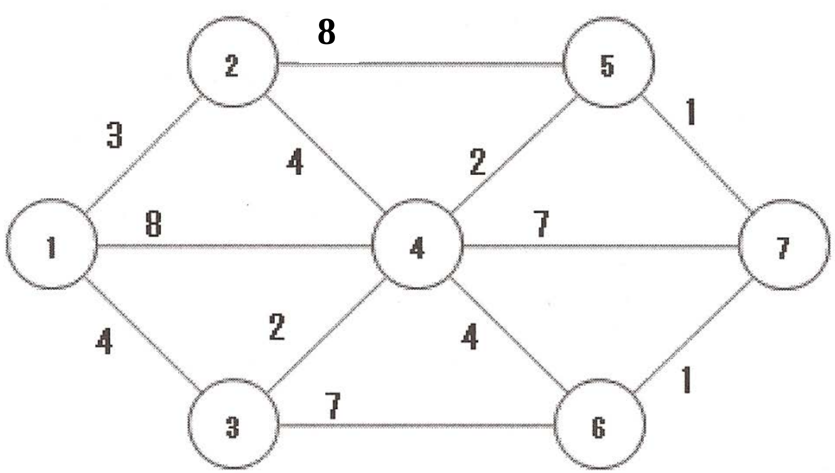
  for $j = 1,\ldots,n$; $j \neq k$ do

  $d_{ij} \leftarrow \min\{d_{ij},\ d_{ik} + d_{kj}\}$

任意の2点間の最短路d_{ij} ← min{d_{ij}, d_{ik}＋d_{kj}}

	1	2	3	4	5	6	7
1	0	3	4	8	∞	∞	∞
2	3	0	∞	4	8	∞	∞
3	4	∞	0	2	∞	7	∞
4	8	4	2	0	2	4	7
5	∞	8	∞	2	0	∞	1
6	∞	∞	7	4	∞	0	1
7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
k=1	1	0	3	4	8	∞	∞	∞
	2	3	0	∞	4	8	∞	∞
	3	4	∞	0	2	∞	7	∞
	4	8	4	2	0	2	4	7
	5	∞	8	∞	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0



		1	2	3	4	5	6	7
k=1	1	0	3	4	8	∞	∞	∞
	2	3	0	7	4	8	∞	∞
	3	4	7	0	2	∞	7	∞
	4	8	4	2	0	2	4	7
	5	∞	8	∞	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	8	∞	∞	∞
k=2	2	3	0	7	4	8	∞	∞
	3	4	7	0	2	∞	7	∞
	4	8	4	2	0	2	4	7
	5	∞	8	∞	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	7	11	∞	∞
	2	3	0	7	4	8	∞	∞
k=3	3	4	7	0	2	15	7	∞
	4	7	4	2	0	2	4	7
	5	11	8	15	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	7	11	∞	∞
k=2	2	3	0	7	4	8	∞	∞
	3	4	7	0	2	15	7	∞
	4	7	4	2	0	2	4	7
	5	11	8	15	2	0	∞	1
	6	∞	∞	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	11	11	∞
	2	3	0	7	4	8	14	∞
k=3	3	4	7	0	2	15	7	∞
	4	6	4	2	0	2	4	7
	5	11	8	15	2	0	∞	1
	6	11	14	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	11	11	∞
	2	3	0	7	4	8	14	∞
	3	4	7	0	2	15	7	∞
k=4	4	6	4	2	0	2	4	7
	5	11	8	15	2	0	∞	1
	6	11	14	7	4	∞	0	1
	7	∞	∞	∞	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	13
	2	3	0	6	4	6	8	11
	3	4	6	0	2	4	6	9
	4	6	4	2	0	2	4	7
k=5	5	8	6	4	2	0	6	1
	6	10	8	6	4	6	0	1
	7	13	11	9	7	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	13
	2	3	0	6	4	6	8	11
	3	4	6	0	2	4	6	9
k=4	4	6	4	2	0	2	4	7
	5	8	6	4	2	0	6	1
	6	10	8	6	4	6	0	1
	7	13	11	9	7	1	1	0

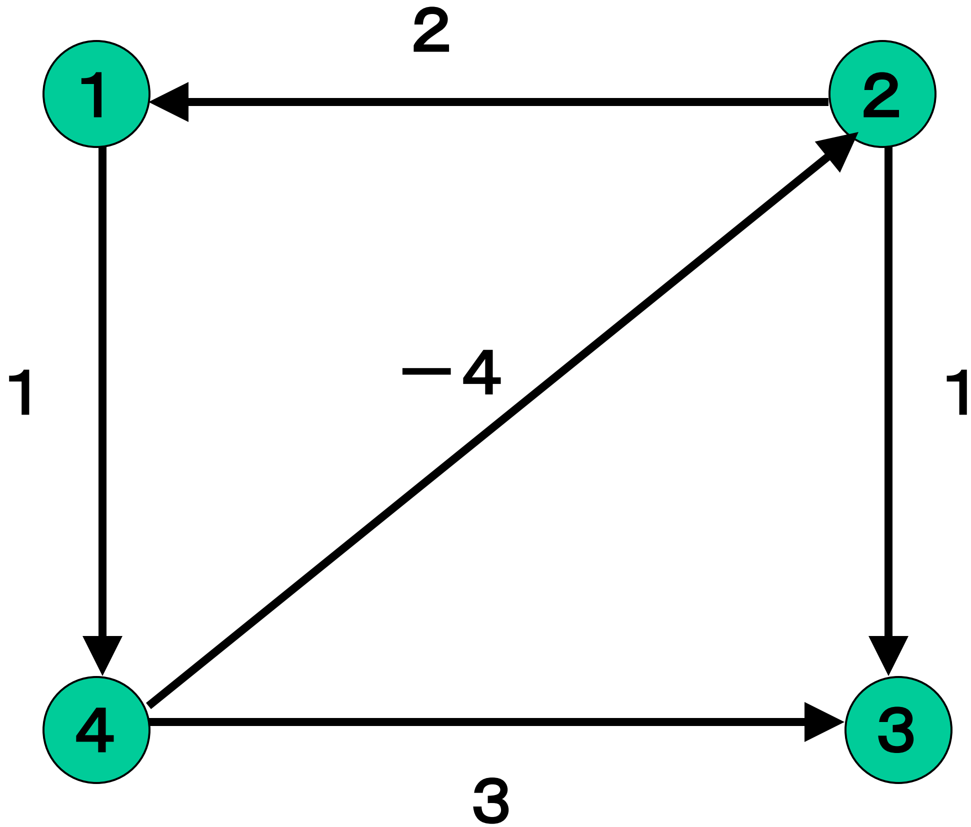
		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	9
	2	3	0	6	4	6	8	7
	3	4	6	0	2	4	6	5
	4	6	4	2	0	2	4	3
k=5	5	8	6	4	2	0	6	1
	6	10	8	6	4	6	0	1
	7	9	7	5	3	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	9
	2	3	0	6	4	6	8	7
	3	4	6	0	2	4	6	5
	4	6	4	2	0	2	4	3
	5	8	6	4	2	0	6	1
k=6	6	10	8	6	4	6	0	1
	7	9	7	5	3	1	1	0

		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	9
	2	3	0	6	4	6	8	7
	3	4	6	0	2	4	6	5
	4	6	4	2	0	2	4	3
	5	8	6	4	2	0	**6**	1
	6	10	8	6	4	**6**	0	1
k=7	7	9	7	5	3	1	1	0

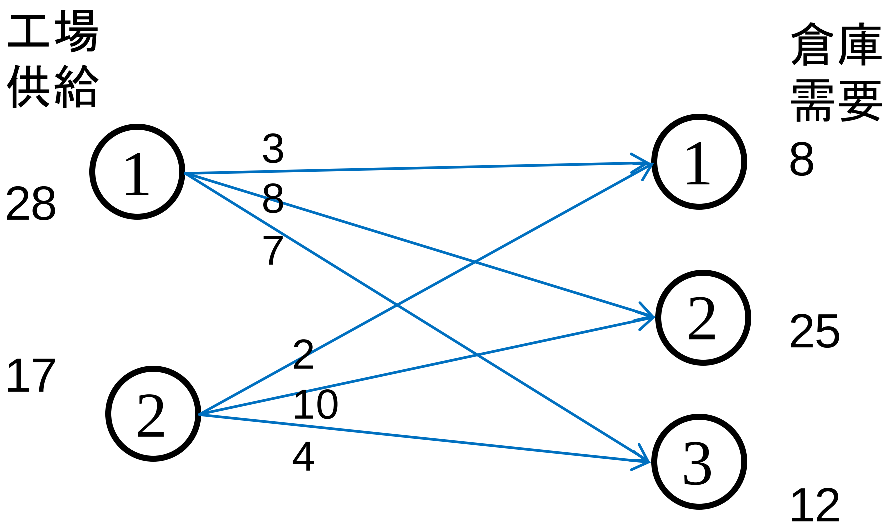
		1	2	3	4	5	6	7
	1	0	3	4	6	8	10	9
	2	3	0	6	4	6	8	7
	3	4	6	0	2	4	6	5
	4	6	4	2	0	2	4	3
	5	8	6	4	2	0	**2**	1
	6	10	8	6	4	**2**	0	1
	7	9	7	5	3	1	1	0

負の閉路を含む最短路問題



輸送問題

- ある製品をm箇所の工場からn箇所の倉庫に最も安い費用で輸送する
- 各工場からの供給可能量: $S_i (i=1,\ldots,m)$
- 各倉庫の需要量: $D_j (j=1,\ldots,n)$
- 工場iから倉庫jへの製品1個あたりの輸送コストをc_{ij}
- 輸送費用を最小化する輸送量 x_{ij} $(i=1,\ldots,m, j=1,\ldots,n)$を求める



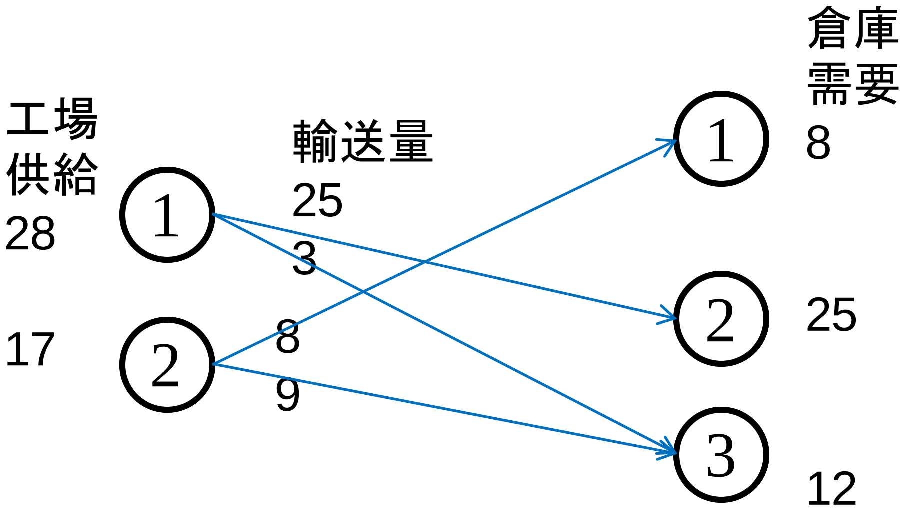
実行可能解設定（ハウザッカー法）
輸送費の安い枝（工場→倉庫）から輸送

工場 ＼ 倉庫	倉庫1		倉庫2		倉庫3		供給量（未達成量）
工場1		3		8		7	28
工場2		2		10		4	17
需要量(未達成量）	8		25		12		

工場 ＼ 倉庫	倉庫1		倉庫2		倉庫3		供給量（未達成量）
工場1		3		8		7	28
工場2		2		10		4	17
需要量(未達成量）	8		25		12		

輸送費用は＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿．

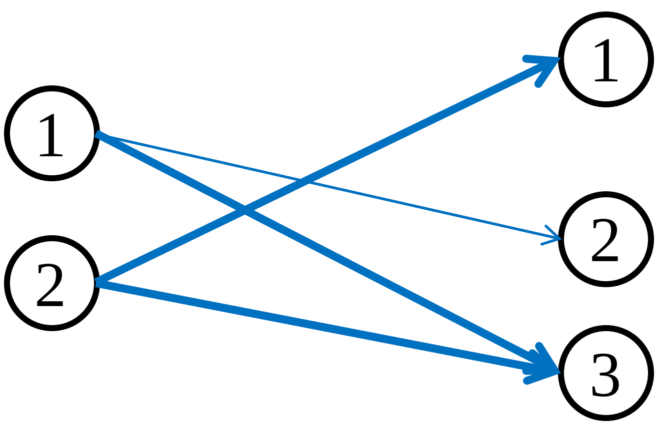
飛び石法による解の改善1



- ハウザッカー法で得られた解を改善
- 輸送量が0の枝(　　)に注目

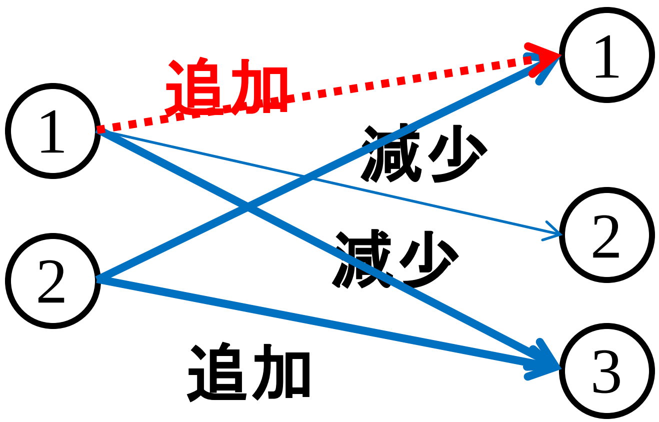
（費用　　で使われていない枝の中で最も安い）

- 輸送量は増加可能か？

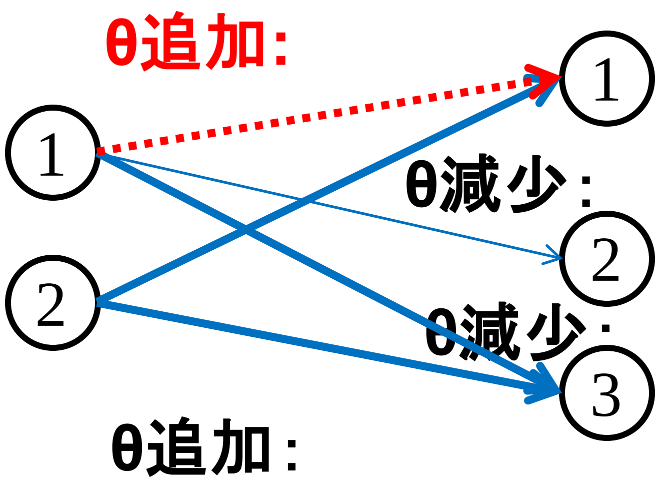


- <u>輸送量が　　の枝</u>を1つ選ぶと、輸送量が正の枝と閉路をなす
- 閉路(ループ)：ある点から同じ点に戻る輸送経路

飛び石法による解の改善2



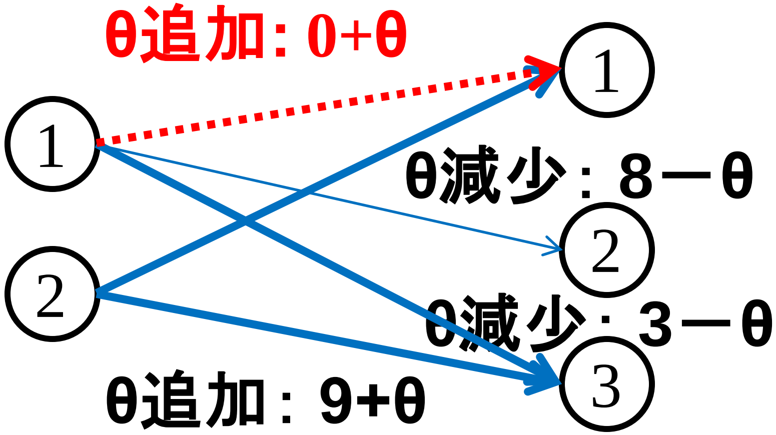
- θループとは何か？
- 輸送量の変動量をθとする
- θループ：閉路上では
輸送量追加と減少の枝が交互に
- 各工場と倉庫には、輸送量追加と
減少の枝が1本づつ接続
- 各工場と倉庫の需要と供給を満足



- 輸送量が0の枝に輸送量θ≧0を追加
- θループ：閉路上では
輸送量θ追加とθ減少の枝が交互に
- 輸送費用の変動を計算

- 枝　　　　に輸送量θを増加させると
輸送費用は　　　　　減少する

飛び石法による解の改善3



- 輸送量が0の枝に輸送量θ≧0を追加
- θループ: 閉路上では
  θ追加−θ減少の枝が交互に

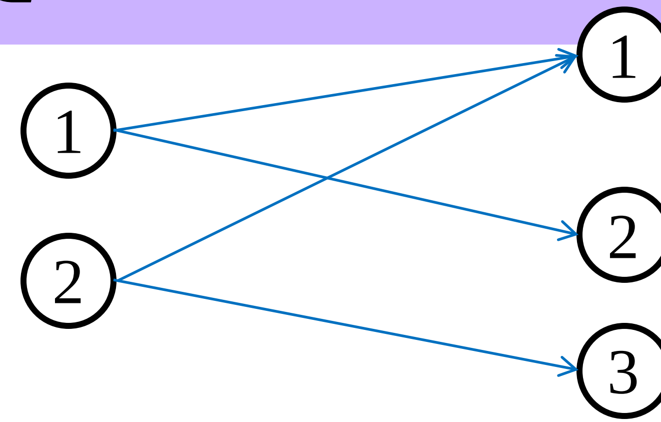
	倉庫1		倉庫2		倉庫3	
工場1	0+θ 増加	3	25	8	3−θ 減少	7
工場2	8−θ 減少	2	0	10	9+θ 増加	4

- 追加輸送量θ≧0は
  - ≧0
  - ≧0
  - ≧0
  - ≧0
  - を満たす最大のθ
- 輸送量θ= を追加または減少
- 輸送費は 減少 (273 ⇒)

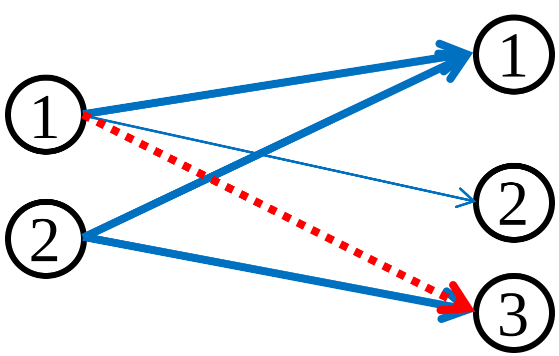
輸送費用は

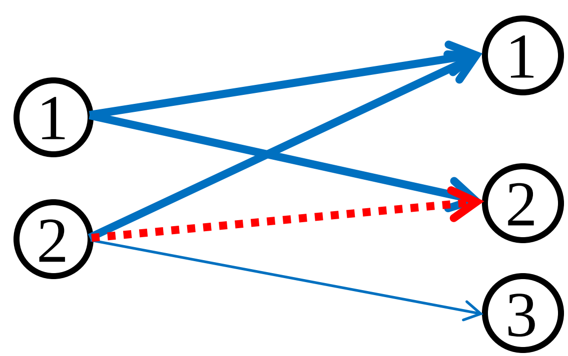
最適性の判定

	倉庫1		倉庫2		倉庫3	
工場1	3	3	25	8	0	7
	増加				減少	
工場2	5	2	0	10	12	4
	減少				増加	



- θループ：費用の改善度を計算⇒改善できない場合（最適解が得られた）
- 輸送量が0である残りの6ー4本の枝に輸送量θを追加できるかチェック
- (1)経路13をθ増加
- (2)経路22をθ増加





施設配置問題

　関東地方を中心に営業を行ってきた輸入品販売業のK社では、関西地方に活動を拡大するため、京阪神地方に倉庫の賃借を行う計画を立てている。賃借の候補となる倉庫はmカ所にあって、第i地点の倉庫W_i $(i=1,\cdots,m)$の月間処理能力はa_i(トン／月)で、その経費(賃借料や維持費など毎月の固定費)はd_i(千円／月)である。また、関西一円に広がる消費地D_j $(j=1,\cdots,n)$での輸入品の需要量b_j(トン／月)と、W_iからD_jへのトン当たり輸送費c_{ij}(千円)が与えられたとして、すべての需要を満たし、毎月の総費用(倉庫経費＋輸送費)を最小にする倉庫配置と輸送計画を求めたい。この問題を数理計画問題として定式化せよ。

施設配置問題（p.100）

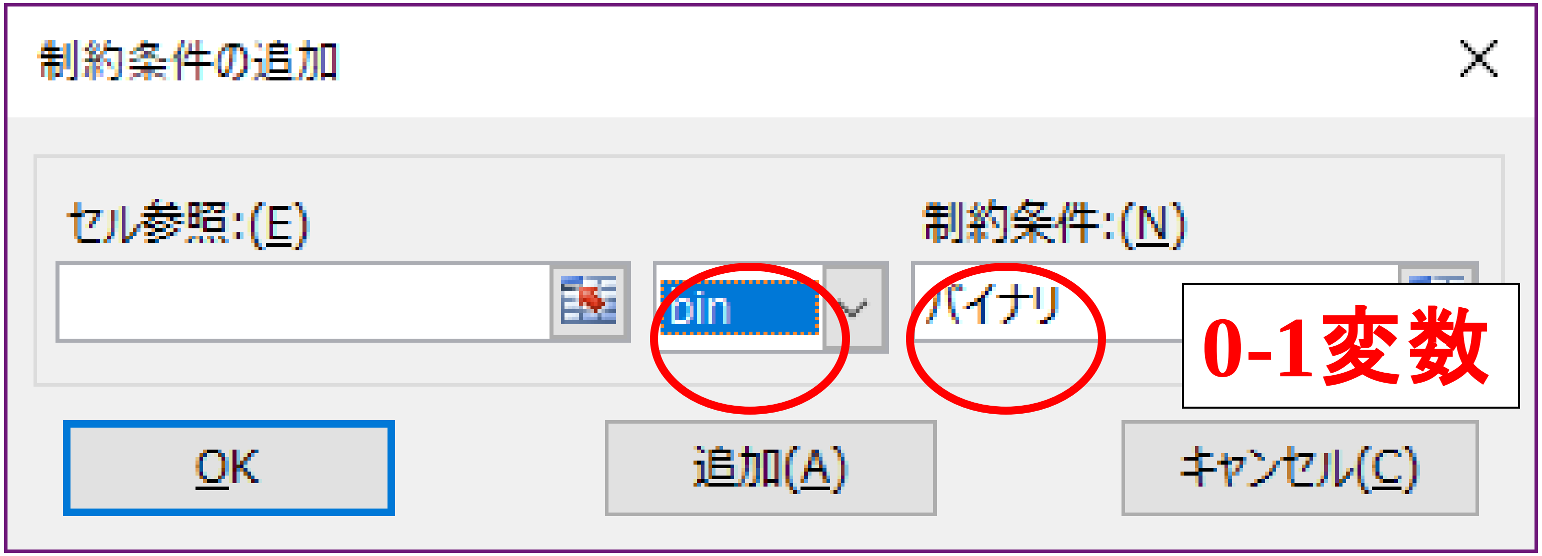
	需要地１	需要地２	需要地３	需要地４	需要地５	処理能力	固定費
倉庫１	4	9	5	10	5	120	120
倉庫２	7	10	3	4	12	110	110
倉庫３	10	11	4	6	2	80	80
倉庫４	9	5	13	5	10	60	60
倉庫５	4	6	12	9	3	60	60
需要	25	30	40	40	50		

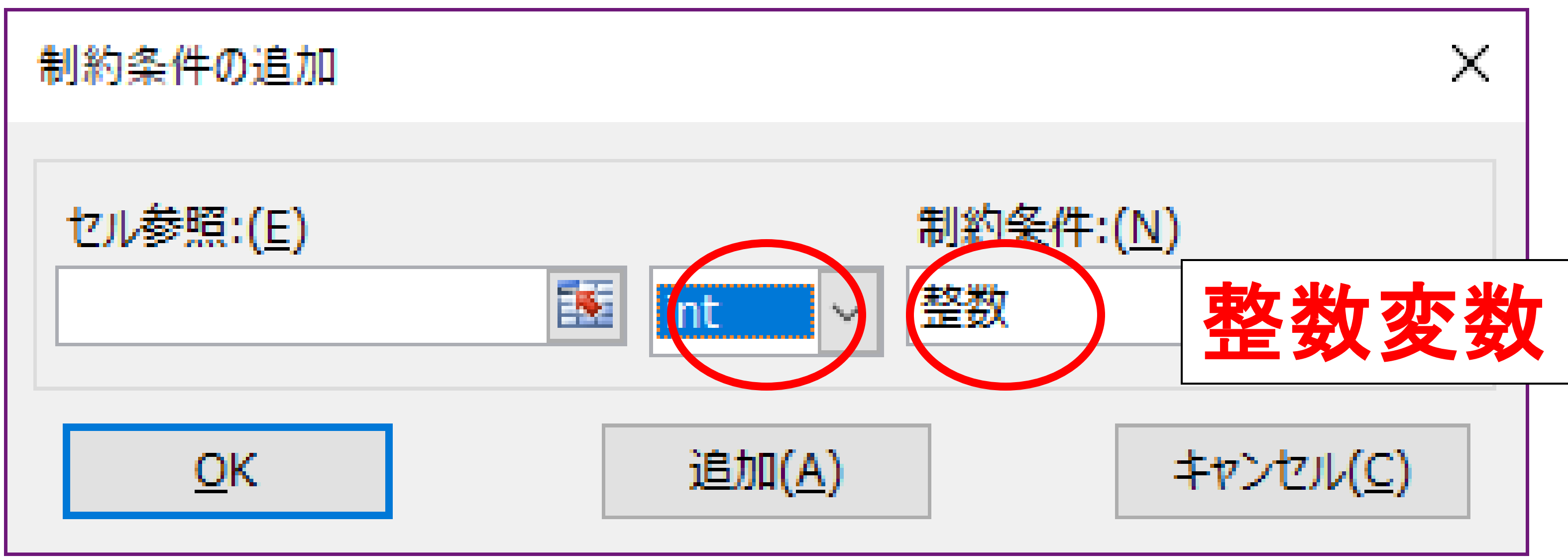
施設配置問題の定式化

- 定数：d_i ＝ 倉庫iの経費（固定費）
  c_{ij} ＝ 倉庫iから需要地jへの輸送単価
  a_i ＝ 倉庫iを借りた場合の供給能力
  b_j ＝ 需要地jの需要量
- 変数：y_i ＝ 倉庫iを借りるとき1、さもなくば0
  x_{ij} ＝ 倉庫iからjへの輸送量（$x_{ij} \geqq 0$）
- 目的関数（総費用最小）：
  最小化 $z = \Sigma_i\ d_i y_i + \Sigma_i \Sigma_j\ c_{ij} x_{ij}$
  制約
  需要を満足しなければならない
  倉庫を借りないならば、送り出してはいけない

変数の0－1制約、整数制約指定

EXCELソルバーでは制約条件の一部として指定





０－１変数を含む
ソルバーパラメータ設定