2019年度
オペレーションズ・リサーチ基礎

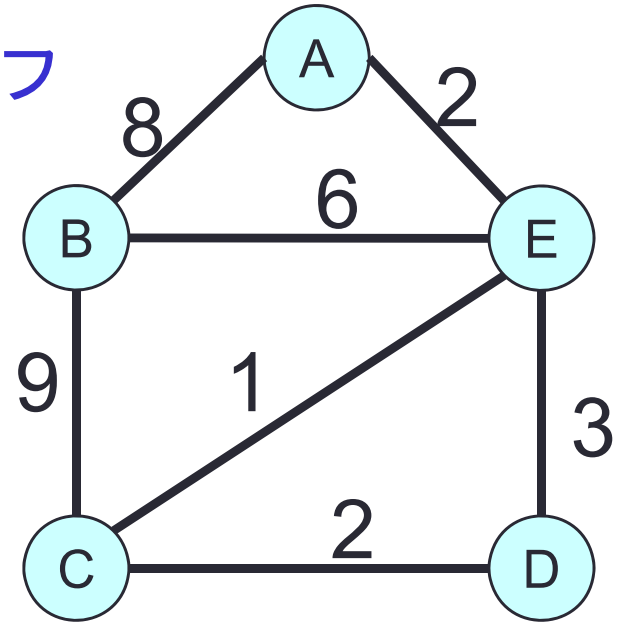
最短路問題
枝長が非負の場合

塩浦昭義
東京工業大学 経営工学系 准教授
shioura.a.aa@m.titech.ac.jp

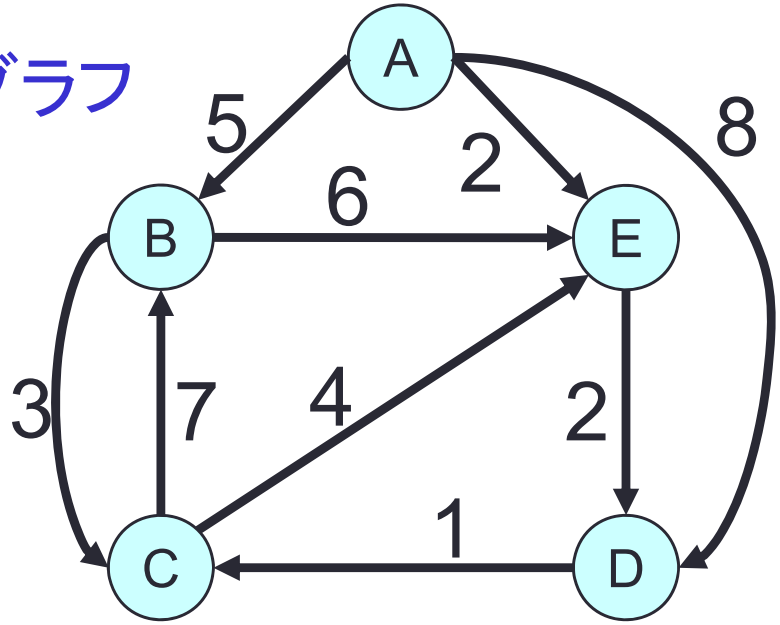
ネットワーク計画問題

- （無向，有向）グラフ
  - 頂点（vertex，接点，点）が枝（edge，辺，線）で結ばれたもの
- ネットワーク
  - 頂点や枝に数値データ（距離，コストなど）が付加されたもの
- ネットワーク最適化問題
  - ネットワークを使って表現される数理計画問題

無向グラフ



有向グラフ



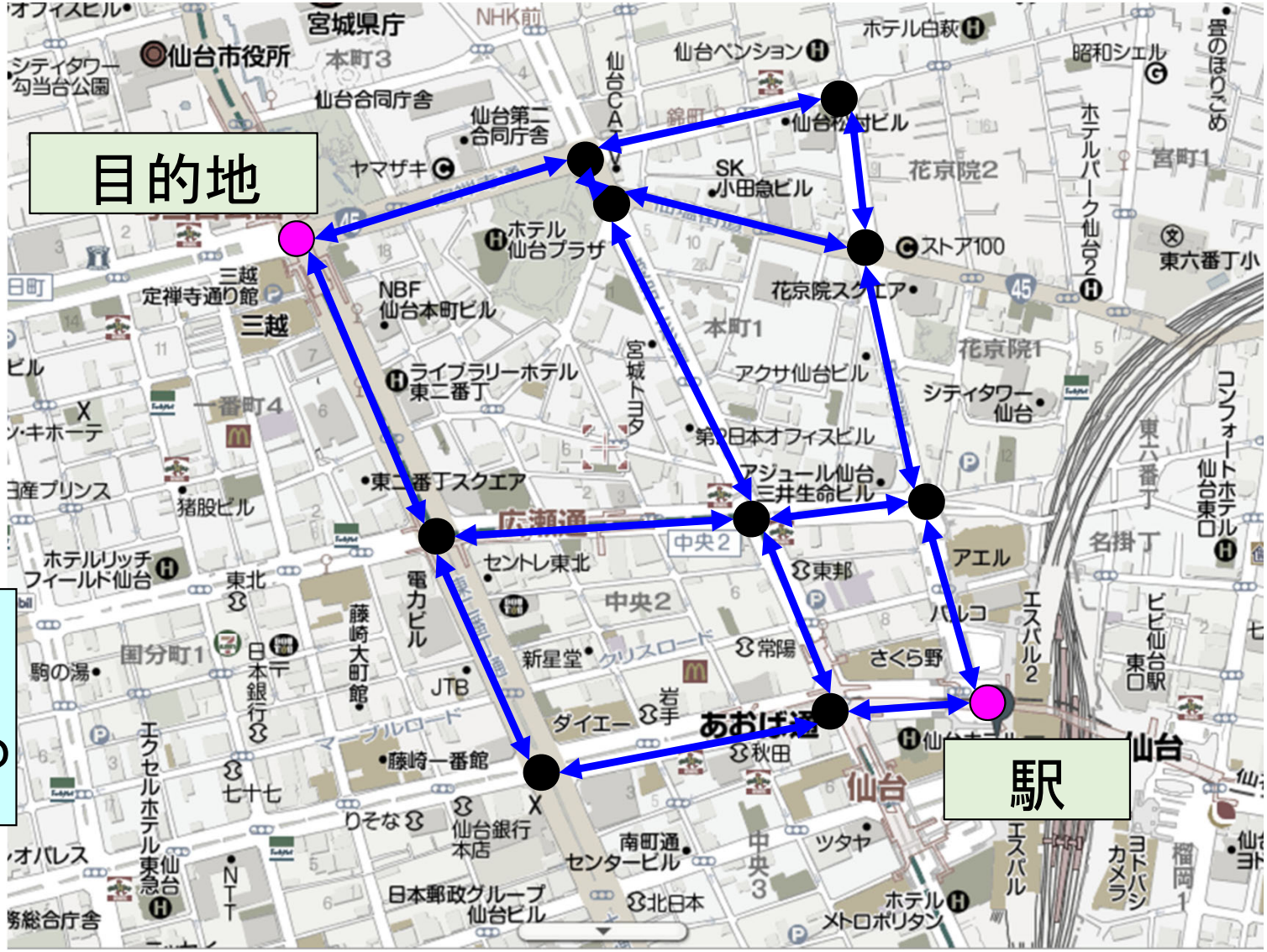
ネットワーク最適化問題の例

「ネットワーク」に関する数理計画問題

最短路問題
最大マッチング問題
最小費用マッチング問題
輸送問題
最大フロー問題
最小費用フロー問題
最小全域木問題
などなど

最も短い経路を求める

駅から
勾当台までの
最短経路を
求めたい
➔グラフを使って
　モデル化



各頂点の間に距離
（移動時間）を与える

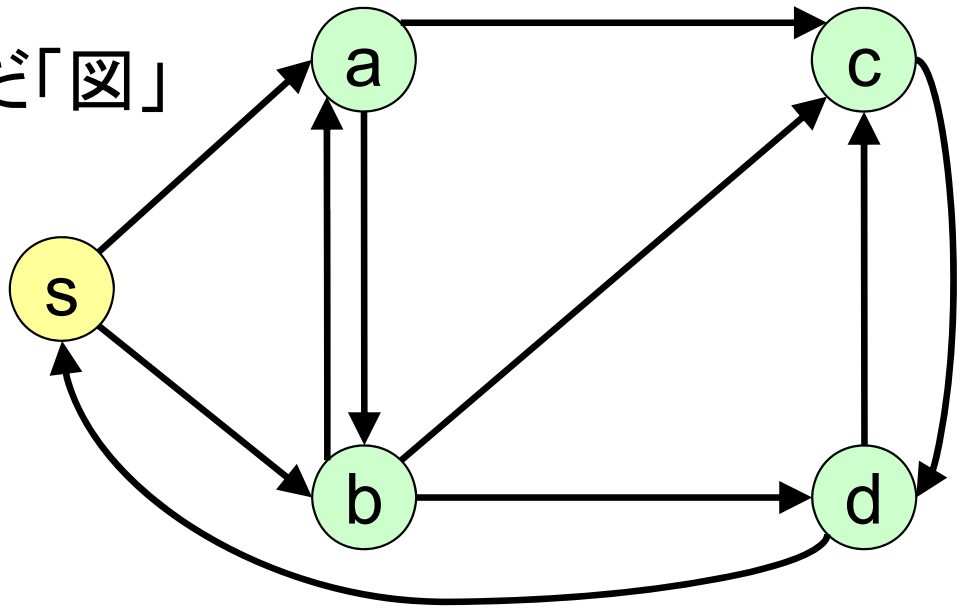
グラフ

最短路問題を数学的に表現するために使う

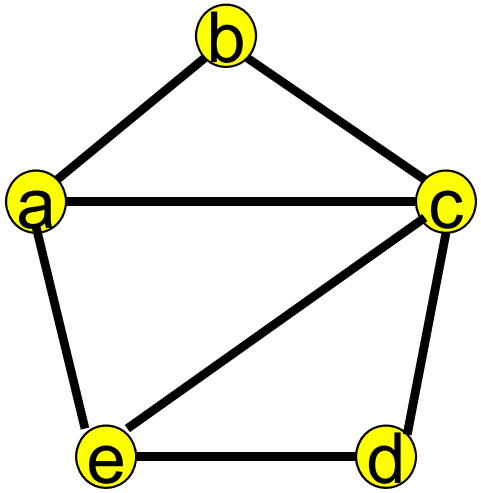
- 定義：グラフ＝「丸」を「線」で結んだ「図」
  - 頂点＝「丸」，枝＝「線」

グラフの例

- 友人関係の図
- 鉄道路線図，道路網
- 組織図，家系図



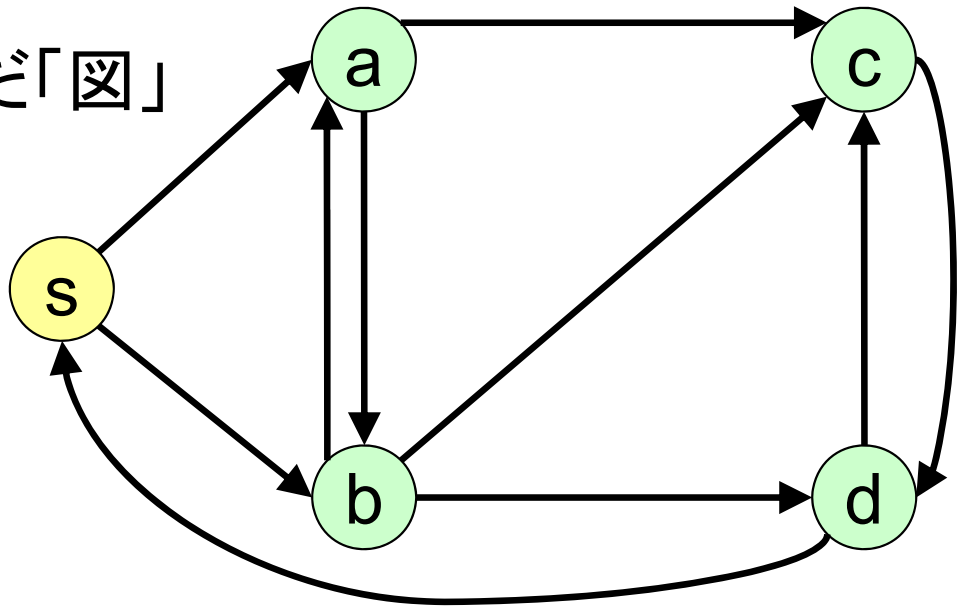
有向グラフ：枝に向きの付いたグラフ
無向グラフ：枝の向きの付いていないグラフ



グラフ

最短路問題を数学的に表現するために使う

- 定義：グラフ＝「丸」を「線」で結んだ「図」
  - 頂点＝「丸」，枝＝「線」

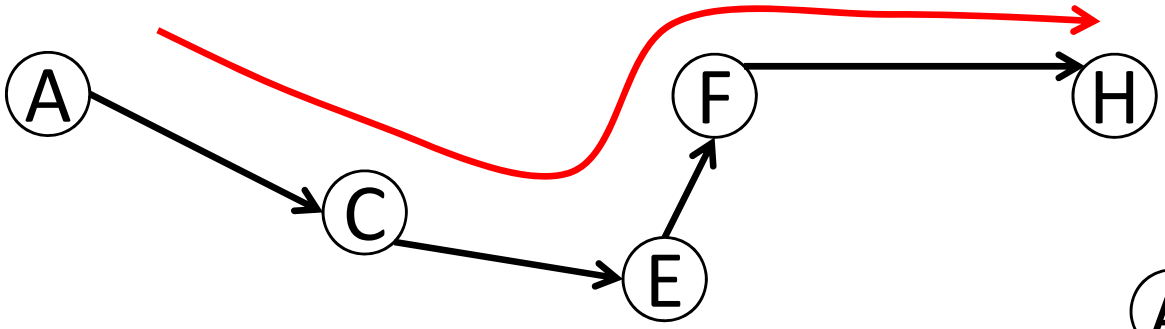


※数学的には，グラフ$G = (V, E)$は
全頂点の集合V と全枝の集合E の対として表現
各枝$(u, v) \in E$は頂点の（順序）対として表現

上記のグラフの場合，
すべての頂点の集合$V = \{s, a, b, c, d\}$
すべての枝の集合$E = \{(s, a), (a, b), (b, a), (b, d), \ldots\}$

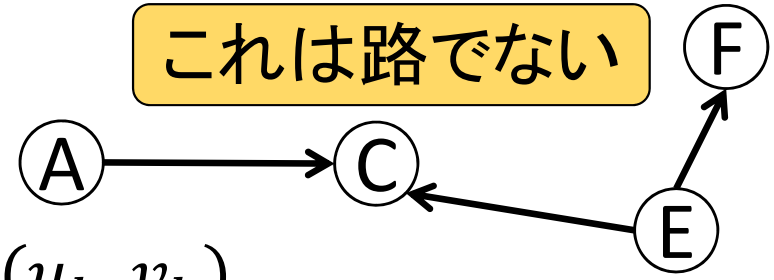
路と閉路

- 路（みち）（path, パス）：複数の枝が1つにつながったもの



枝の向きに沿って進む

これは路でない



厳密な定義：枝の列 $(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$

ただし $v_1 = u_2, v_2 = u_3, \ldots, v_{k-1} = u_k$

注意：路・閉路は，同じ頂点，
同じ枝を何回通っても可

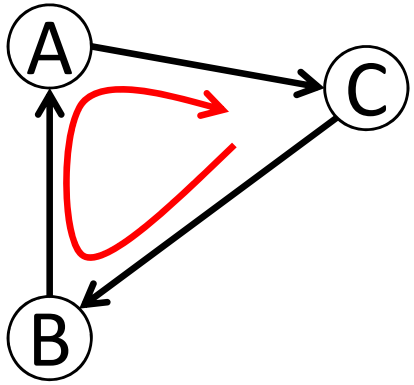
- 閉路（cycle, サイクル）：

複数の枝が1つの輪になったもの

厳密な定義：枝の列 $(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$

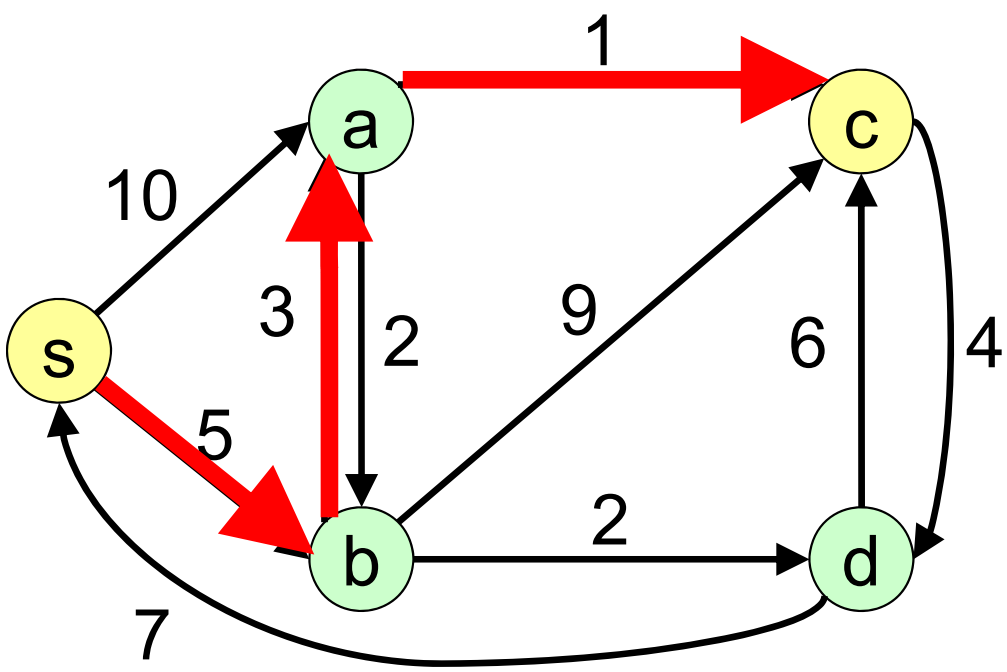
ただし $v_1 = u_2, v_2 = u_3, \ldots, v_{k-1} = u_k$

$v_k = u_1$



単一始点単一終点 最短路問題

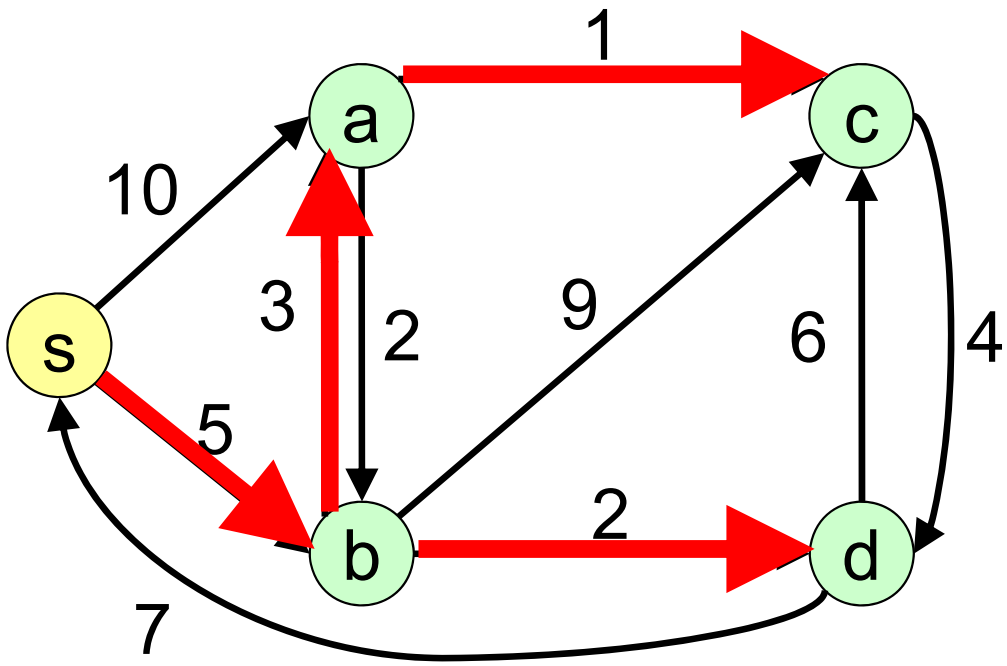
- 入力：有向グラフ G=(V, E)
  各枝の長さ $\ell(e) \geq 0$ $(e \in E)$, 始点 s∈V，終点t∈V
- 出力：s から t への最短路 P(t) とその長さd(t)
  （sからt への最短路 P*
  = sからtへの路のうち，枝の長さの和 $\ell(P^*)$が最小のもの）



後述するように，
特定の頂点への最短路を
求めるためには，
全ての頂点への最短路を
求めると効率的
➔単一始点全終点
最短路問題

単一始点全終点 最短路問題

- 入力：有向グラフ G=(V, E)
  各枝の長さ $\ell(e) \geq 0$ $(e \in E)$, 始点 s∈V
- 出力：s から すべての頂点 v への最短路P(v)とその長さd(v)



以降で使う仮定：
s から各頂点 v への路が存在
（存在しない場合は
枝(s,v)を追加，その長さを
十分大きい正数とする）

最短路の性質

最短路の存在

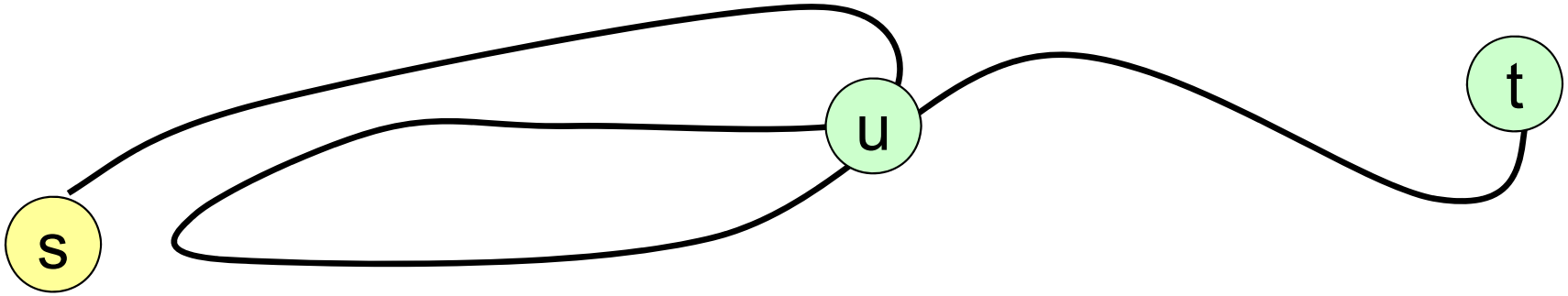
命題: 枝長が全て非負
➔ 任意の頂点 s, t に対し，s から t への最短路が存在

考え方:

- 同じ頂点を2回以上通ることを許せば，s から t の路の数は無限個存在しうる ← 最短路の存在は非自明
- 一方，各頂点を高々1回通る路は有限個 ← 最短路の存在は自明

∴ ある頂点u を2回以上通る路 P に対し，次の路 Q が存在することを示せば良い:

- Q の長さ ≦ P の長さ
- 2回以上通過する頂点の個数: P より Q の方が少ない

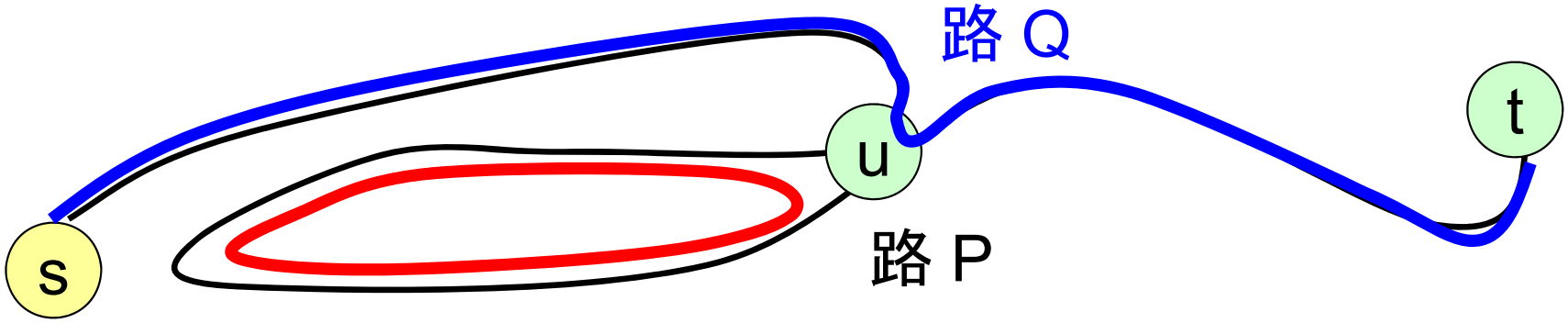


最短路の存在（つづき）

∴ ある頂点u を2回以上通る路 P に対し，
次の路 Q が存在することを示せば良い：

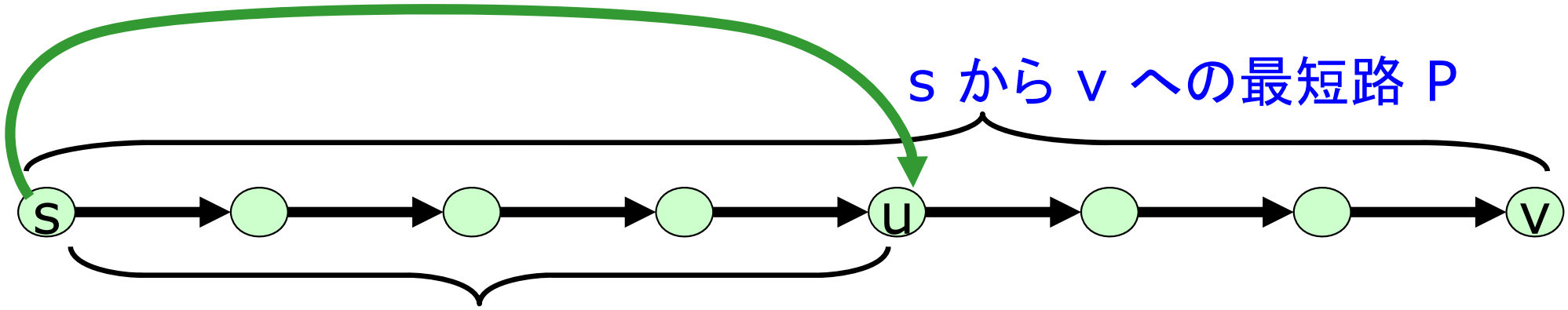
- Q の長さ ≦ P の長さ
- 2回以上通過する頂点の個数：P より Q の方が少ない

路 P において，最初に u が現れてから最後に出てくるまでの
部分路（閉路）を削除 ➔ 路 Q
削除した部分の枝長 ≧ 0　∴ 路 Q の長さ ≦ 路 P の長さ
頂点 u の通過回数：2以上から1に減少
他の頂点の通過回数：以前より増えない
∴ 2回以上通過する頂点の個数：P より Q の方が少ない　■



最短路の部分路は最短路

命題　P: 頂点 s から頂点 v への最短路
　P は途中に頂点 u を含むと仮定
　➔ s から u への P の部分路 P' は，s から u への最短路



s から u への部分路 P′ ← s から u への最短路

（証明）もし P′ より短い路 P″ が存在したら
　→sからvへの路として，まず P″ に沿って s から u に行き，
　　その後 P と同じ枝をたどって v に行く路 Q を考える.
　Q の長さ - P の長さ ＝ P″ の長さ - P′ の長さ ＜ 0
　　よって，P の選び方に矛盾.

ダイクストラ法による最短路長の計算（枝長が非負の場合）

最短路長の性質

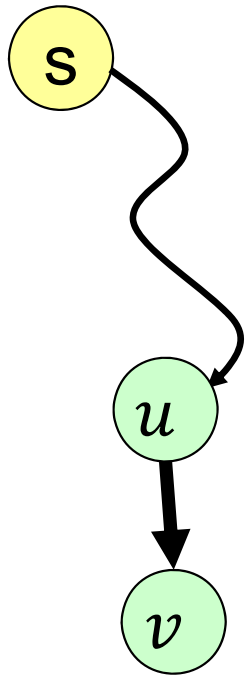
- 定義: d(v) = 頂点 s から v への最短路長

- 枝長≧0 ➔ d(v)≧0 (∀v∈V)
- s から s への最短路長 d(s) = 0 ∴全ての最短路の中で最も短い

- 任意の v ≠ s に対し，ある頂点 u が存在して，
  s から v への最短路 = s から u への最短路 + (u,v)

∴ d(v) = d(u) + l(u,v) ≧ d(u)

∴最短路長は，小さい順に計算すると都合が良い



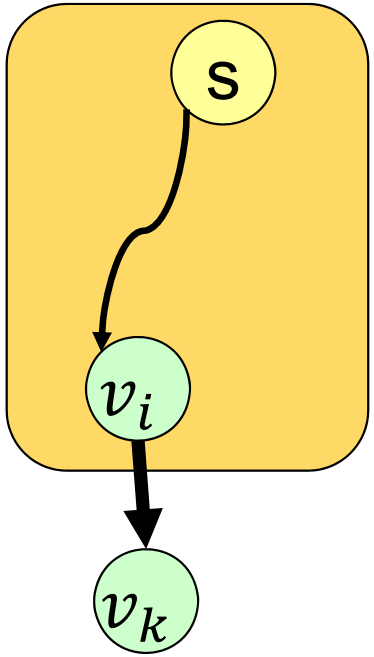
ダイクストラ法：考え方

各頂点への最短路（の長さ）を，d(v) の小さい方から順に計算する
（長さが同じ ➔ 最短路の枝数の少ない順に）

最短路を，小さい方から k-1 番目まで計算したと仮定

定義： v_i ＝ i 番目に短い最短路の行き先

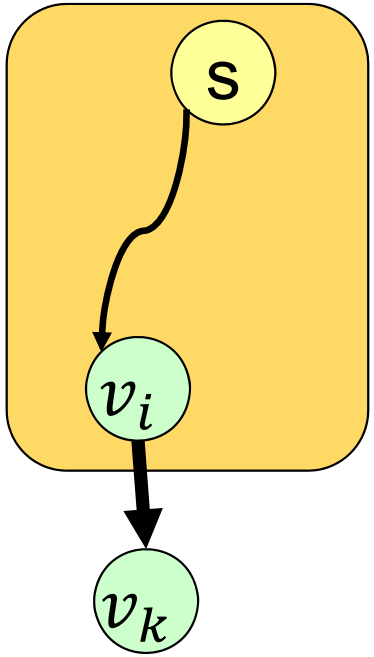
$$X = \{v_1, v_2, \dots, v_{k-1}\}$$



k 番目に短い最短路 P_k と
その行き先 $v_k \in V - X$を計算したい

ダイクストラ法：考え方（つづき）

k 番目に短い最短路 P_k と
　　その行き先 $v_k \in V - X$を計算したい



- ある頂点 u が存在して，
　s から v_k への最短路 = s から u への最短路 + (u, v_k)
- d(u)≦d(v_k)　➔ u∈X ➔ u = v_i ($\exists\, v_i \in X$)

∴ ある $v_i \in X$ が存在して，

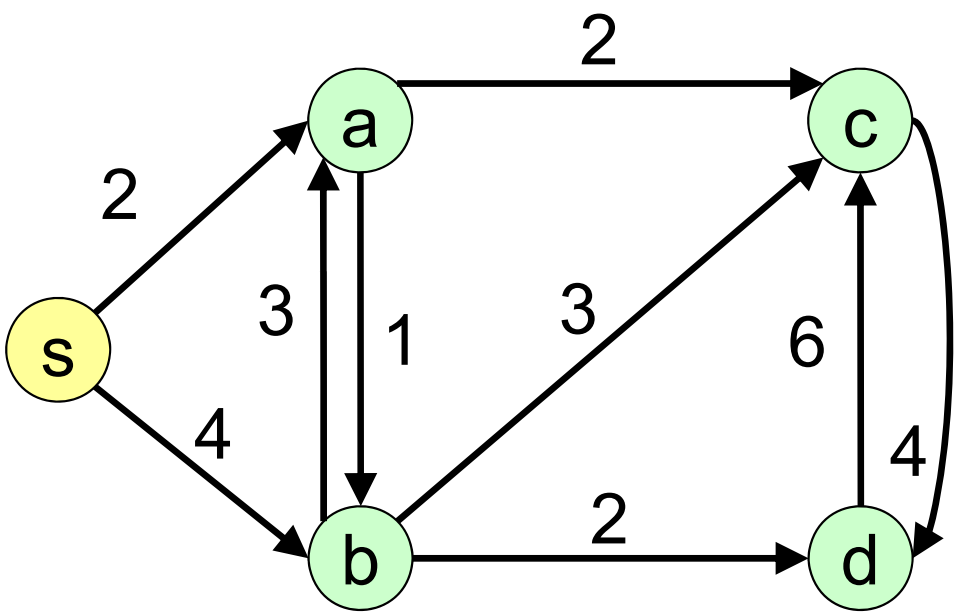
P_k = s から v_i への最短路 + (v_i, v_k)

このような路の長さが最も短くなるように，

$v_i \in X$ および X の内外を結ぶ枝 (v_i, v) を選ぶ ➔ $v_k = v$ となる

以上の計算を効率的に行う ➔ ダイクストラ法

k 番目に短い最短路の計算



v_1 = s, d(s)=0

v_2 = a, d(a)=2

v_3 = b, d(b)=3

4番目に短い最短路
およびその行き先v_4 を求めたい

候補:
① a への最短路 + 枝 (a,c): 長さ 2+2=4
② b への最短路 + 枝 (b,c): 長さ 3+3=6
③ b への最短路 + 枝 (b,d): 長さ 3+2=5

①が最小 ➔ 4番目の最短路，v_4 = c

ダイクストラのアルゴリズム

ステップ0: $d(s) = 0, d(v) = +\infty\ (\forall v \neq s)$ とおく.

X={s} とおき，s から出る各枝 (s,v) に対し $d(v) = \ell(s, v)$ とおく

ステップ1: V-X の要素 v で d(v) 最小の v=v* を選び，Xに追加.

ステップ2: $X = V$ ➔ 各v に対して d(v) を最短路長として出力，終了.

ステップ3: v* から出る各枝 (v*,v) に対し，

$v \in V - X$ かつ$d(v) > d(v^*) + \ell(v^*, v)$ ならば，

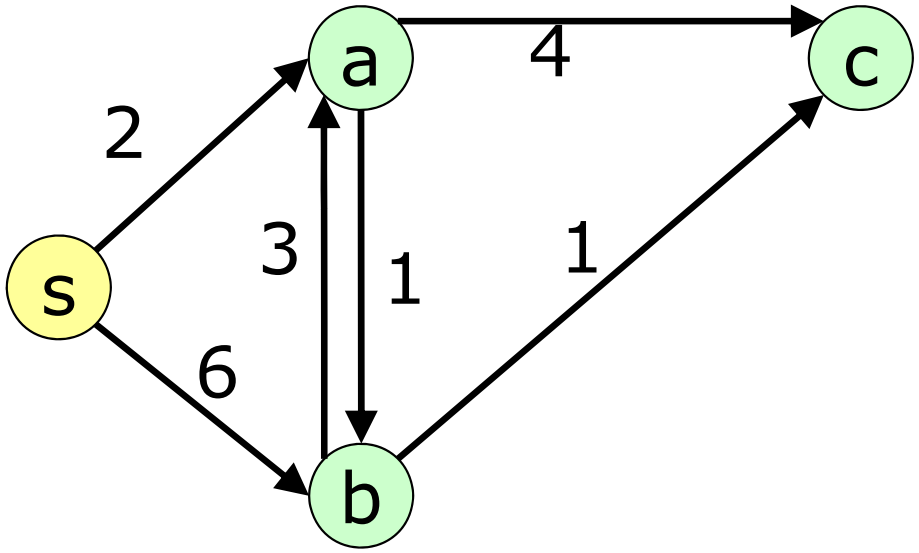
$d(v) := d(v^*) + \ell(v^*, v)$ とおく.

ステップ1に戻る.

定理 枝長が非負のグラフにおいて，
ダイクストラのアルゴリズムは始点 s から各頂点への
最短路長を正しく計算する.

最短路長だけでなく，最短路を計算することも
（若干の修正により）可能である

実行例



	k=0	1	2	3
s	0			
a	∞	2		
b	∞	6	3	
c	∞	∞	6	4

枝長の非負性について

- ダイクストラ法は枝長が非負の場合，最短路長を正しく計算
- 枝長が負の枝をもつばあいは？　反例は？
- これまでの議論のうち，枝長が負の場合はどこがおかしくなる？