

最適化手法 第 8 回
ネットワーク最適化 (2)
— 最短路問題 —

後藤 順哉
中央大学 理工学部 経営システム工学科

2014 年 5 月 28 日

今日の概要

今日の目標

- 最短路問題の定義と解法を理解する
- 最短路問題を解けるようになる

ことわり

本講義で用いる資料は、電気通信大学 岡本吉央先生が 2013 年度前期「最適化手法」で用いた資料を、後藤が適宜変更して用いています。

目次

- 1 最短路問題とは？
- 2 最短路の重要な性質
- 3 辺長がすべて同じ場合：幅優先探索
- 4 一般の場合：Dijkstra 法
- 5 今日のまとめと今後の予告

最短路問題とは？

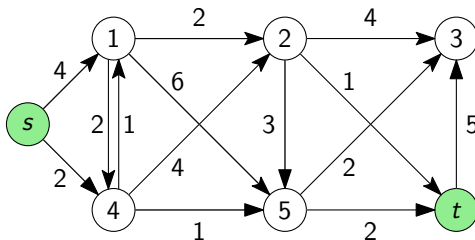
最短路問題とは？

入力

- 有向グラフ $G = (V, E)$, 各辺 $e \in E$ の長さ, 2 頂点 $s, t \in V$

出力

- s を始点とし, t を終点とする G の道で, そこに現れる辺の長さの和が最小のもの



最短路問題とは？

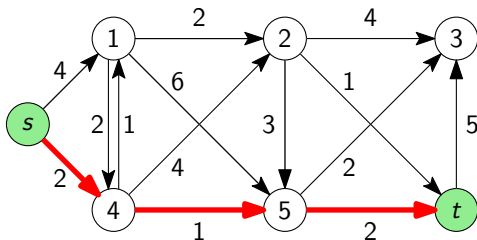
最短路問題とは？

入力

- 有向グラフ $G = (V, E)$, 各辺 $e \in E$ の長さ $\ell(e)$, 2 頂点 $s, t \in V$

出力

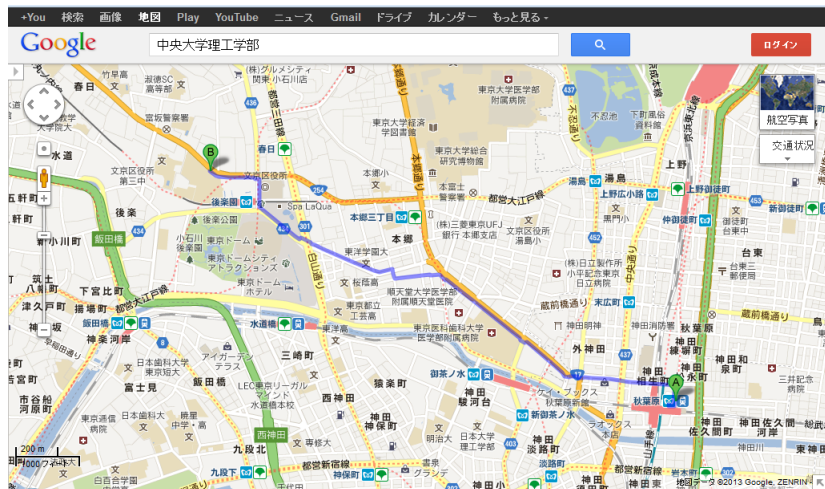
- s を始点とし, t を終点とする G の道で, そこに現れる辺の長さの和が最小のもの



$$\text{辺長和} = 2 + 1 + 2 = 5$$

最短路問題が現れる場面：経路探索 (1)

例えば, Google Maps



最短路問題が現れる場面：経路探索 (2)

旧藤澤研の経路探索サーバ



頂点数 264,346, 辺数 733,846

最短路問題が現れる場面 (2) : 乗車券の分割購入

京葉線で、東京駅から蘇我駅まで最も安く行くには？



注：最短路問題における「長さ」は比喻

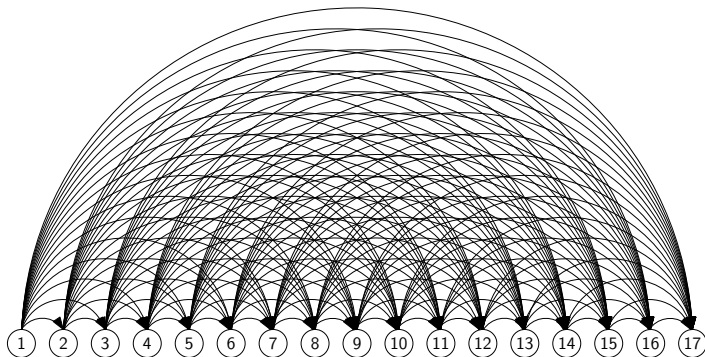
最短路問題が現れる場面 (2)：乗車券の分割購入

京葉線で、東京駅から蘇我駅まで最も安く行くには？

運賃(円)	東京	八丁堀	越中島	潮見	新木場	葛西臨海公園	舞浜	新浦安	市川塩浜	二俣新町	南船橋	新習志野	海浜幕張	検見川浜	稲毛海岸	千葉みなと	蘇我
東京																	
八丁堀	130																
越中島	130	130															
潮見	150	150	130														
新木場	160	160	150	130													
葛西臨海公園	210	160	160	150	150												
舞浜	210	210	160	160	150	130											
新浦安	290	210	210	210	160	150	150										
市川塩浜	290	290	290	210	210	160	150	130									
二俣新町	380	380	290	290	290	210	160	160	150								
南船橋	450	380	380	380	290	290	210	160	160	150							
新習志野	450	450	450	380	380	290	290	210	210	150	130						
海浜幕張	540	540	450	450	380	380	290	290	210	160	150	150					
検見川浜	540	540	540	450	450	380	380	290	290	210	160	150	130				
稲毛海岸	620	540	540	450	450	380	380	290	290	210	160	160	150	130			
千葉みなと	620	620	620	540	540	450	450	380	380	290	210	210	160	150	150		
蘇我	740	740	740	650	650	570	570	480	400	400	320	230	230	190	190	180	

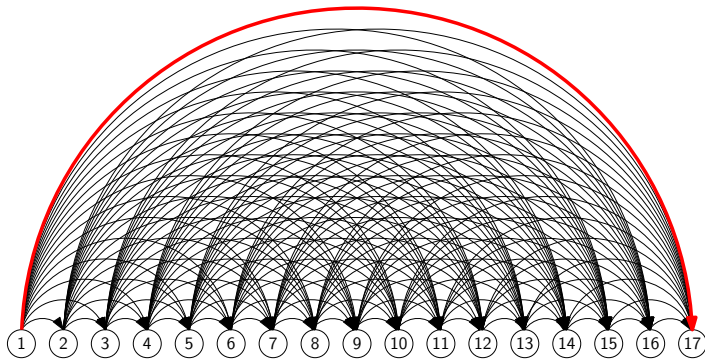
<http://bunkatsu.info/> に基づく

最短路問題が現れる場面 (2)：乗車券の分割購入



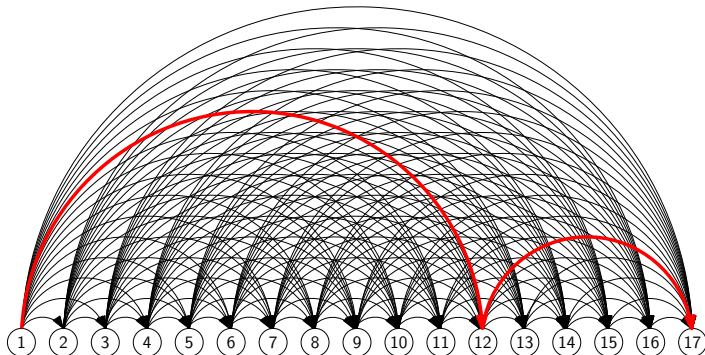
- (1) 東京, (2) 八丁堀, ..., (17) 蘇我
- 頂点 i を始点, 頂点 j を終点とする辺の長さ
= 駅 i から駅 j へ直接行く乗車券の価格

最短路問題が現れる場面 (2)：乗車券の分割購入



東京 → 蘇我：740 円

最短路問題が現れる場面 (2)：乗車券の分割購入



東京 → 新習志野 → 蘇我：450 円 + 230 円 = 680 円 (最適解)

今から行うこと

まず、重要な性質を 1 つ観察

最短路の一部も最短路

次に、最短路問題の解き方を学修

- ① すべての辺の長さが同じ場合
 - アルゴリズム：幅優先探索
- ② そうとは限らない場合
 - アルゴリズム：Dijkstra 法 (ダイクストラ法)

重要な仮定

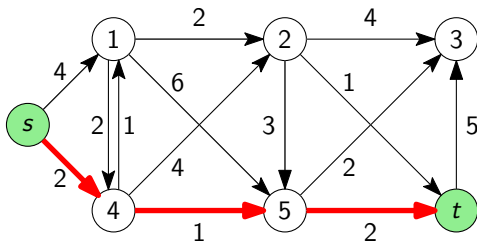
辺の長さはすべて非負 (0 以上)

辺の長さが負となる応用例もあるが、ここでは扱わない

目次

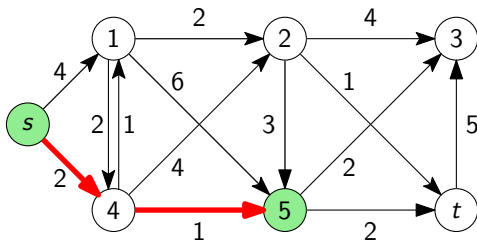
- 1 最短路問題とは？
- 2 最短路の重要な性質
- 3 辺長がすべて同じ場合：幅優先探索
- 4 一般の場合：Dijkstra 法
- 5 今日のまとめと今後の予告

最短路の一部も最短路 (1)



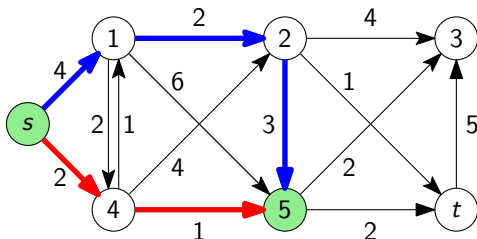
- これは s を始点, t を終点とする最短路
- 頂点 5 を通っている

最短路の一部も最短路 (2)



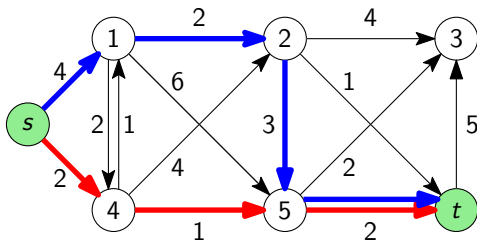
- この道に沿って、 s から 5 まで進むと、それは s を始点、5 を終点とする最短路である
- なぜか？

最短路の一部も最短路 (3)



- そうでないとすると,
 s を始点, 5 を終点とする最短路は別の道である

最短路の一部分も最短路 (4)

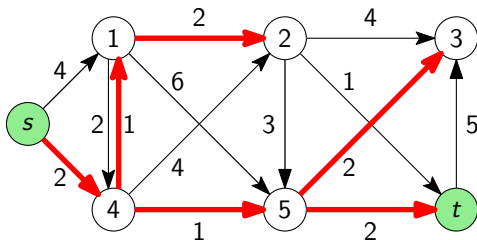


- その道を使うと、 s を始点、 t を終点とする道として始めに考えた最短路よりも短いものが見つかる
- これは、始めに考えた道が最短路であったことに矛盾する

最短路を見つけるための基本アイデア

最短路の見つけ方：基本アイデア

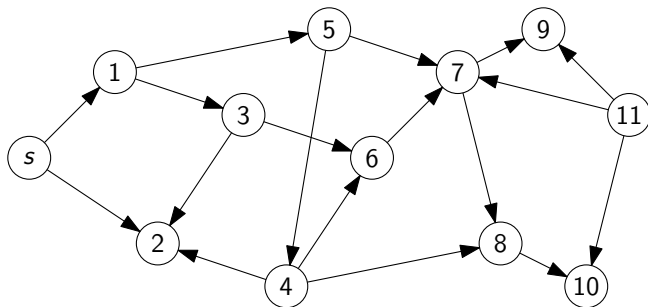
- s を始点, t を終点とする最短路を見つけるだけでなく
 s を始点, 他の各点を終点とする最短路を全部見つける
- 長さが小さい方から順に見つける



目次

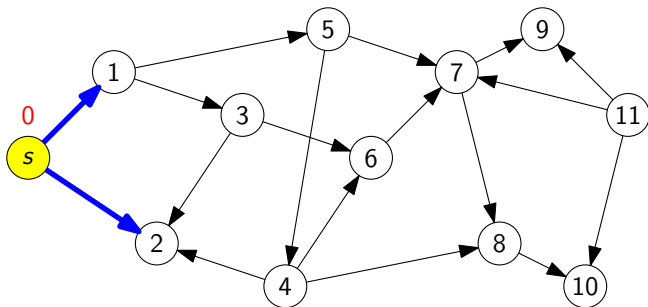
- 1 最短路問題とは？
- 2 最短路の重要な性質
- 3 辺長がすべて同じ場合：幅優先探索
- 4 一般の場合：Dijkstra 法
- 5 今日のまとめと今後の予告

最短路問題：辺長がすべて同じ場合



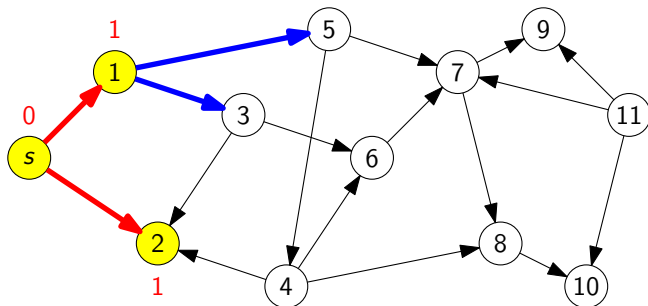
辺長はすべて「1」であるとする

幅優先探索 (breadth-first search, BFS)



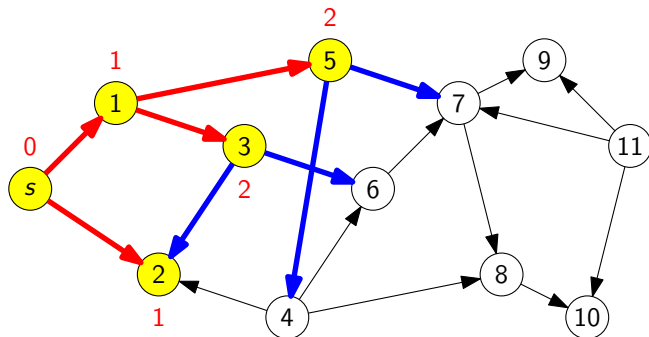
そこを始点とする辺の終点は長さ 1 の道でたどり着ける頂点

幅優先探索 (breadth-first search, BFS)



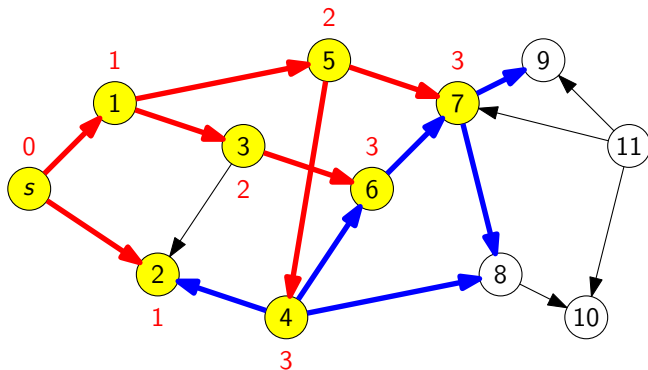
長さ 1 の道でたどり着ける頂点

幅優先探索 (breadth-first search, BFS)



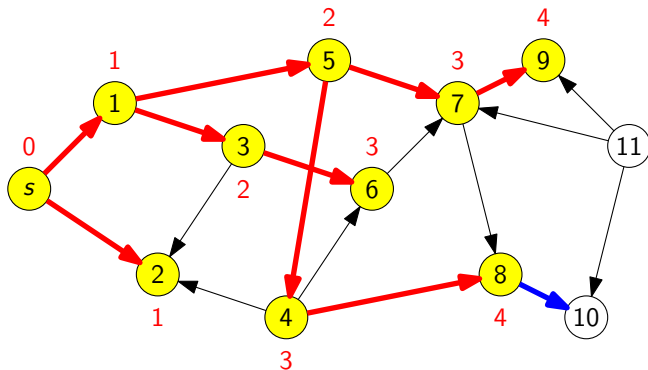
長さ 2 の道でたどり着ける頂点

幅優先探索 (breadth-first search, BFS)



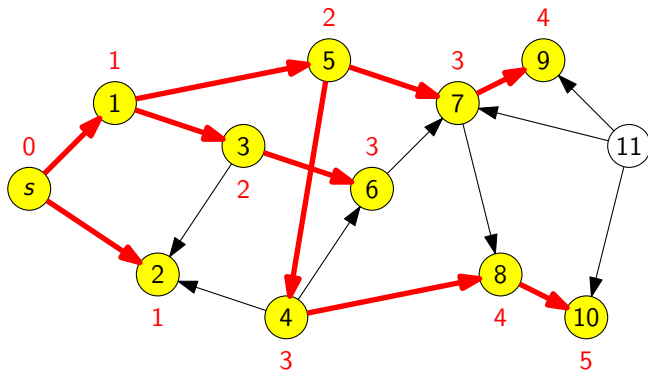
長さ 3 の道でたどり着ける頂点

幅優先探索 (breadth-first search, BFS)



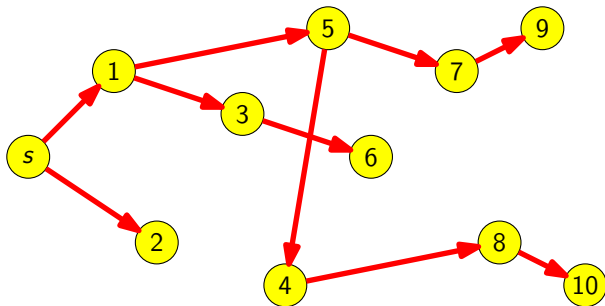
長さ 4 の道でたどり着ける頂点

幅優先探索 (breadth-first search, BFS)



長さ 5 の道でたどり着ける頂点

幅優先探索木 (BFS 木)



幅優先探索木から、最短路を読みとることができる

目次

- 1 最短路問題とは？
- 2 最短路の重要な性質
- 3 辺長がすべて同じ場合：幅優先探索
- 4 一般の場合：Dijkstra 法**
- 5 今日のまとめと今後の予告

Edsger Dijkstra

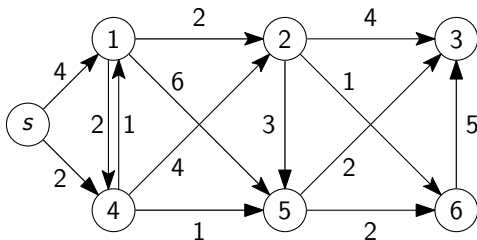


Edsger W. Dijkstra
エドガー・ダイクストラ (1930–2002)

http://en.wikipedia.org/wiki/File:Edsger_Wybe_Dijkstra.jpg

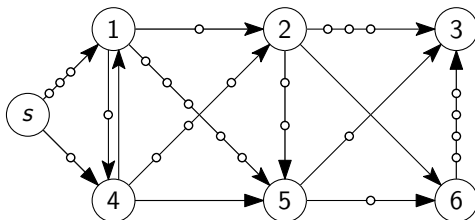
Dijkstra 法 : 基本アイディア

辺を細かくして, 辺長がすべて 1 になるようにする
 ~> 幅優先探索を適用



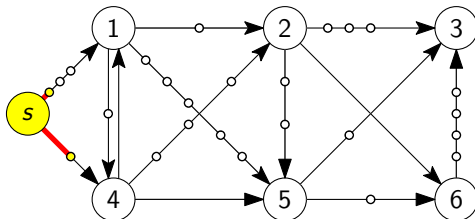
Dijkstra 法 : 基本アイディア

辺を細かくして、辺長がすべて 1 になるようにする
 ~> 幅優先探索を適用



Dijkstra 法 : 基本アイディア

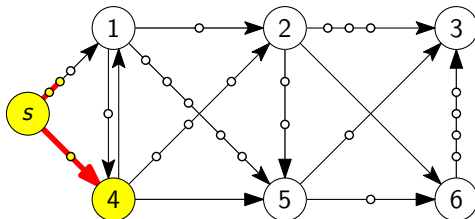
辺を細かくして、辺長がすべて 1 になるようにする
 ⇨ 幅優先探索を適用



長さ 1 の道でたどり着ける頂点

Dijkstra 法 : 基本アイディア

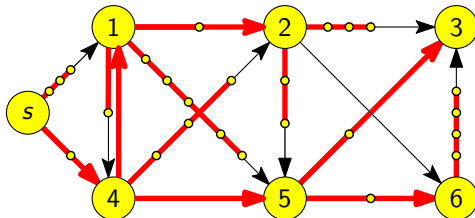
辺を細かくして、辺長がすべて 1 になるようにする
 ~> 幅優先探索を適用



長さ 2 の道でたどり着ける頂点

Dijkstra 法：基本アイデア

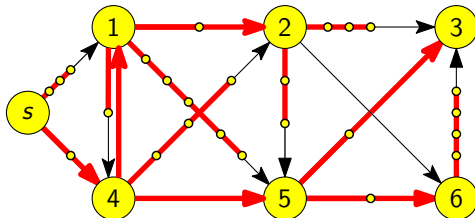
辺を細かくして、辺長がすべて1になるようにする
 ~> 幅優先探索を適用



長さ 9 の道でたどり着ける頂点

Dijkstra 法：基本アイディア

辺を細かくして、辺長がすべて 1 になるようにする
 ~> 幅優先探索を適用

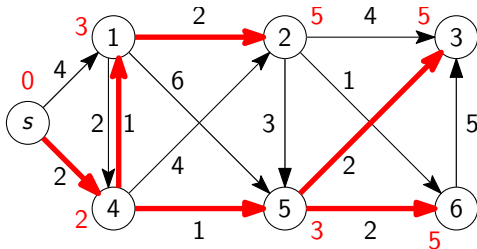


幅優先探索の実行が終了

Dijkstra 法：基本アイディア

辺を細かくして、辺長がすべて 1 になるようにする

⇒ 幅優先探索を適用

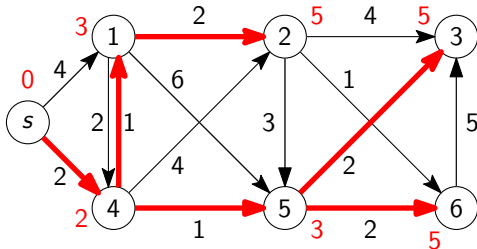


計算された**最短路木** (最短路を読みとることができる)

実際の Dijkstra 法

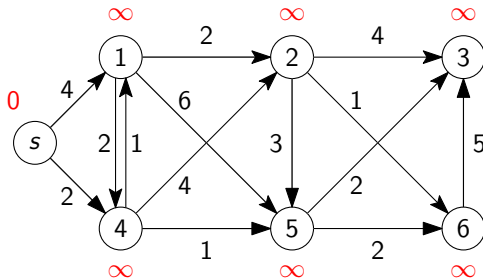
「基本アイデア」の冗長な部分を省く

- ラベルの導入
- ラベルの更新 (幅優先探索のように)
- 確定頂点と未確定頂点 (幅優先探索のように)



実際の Dijkstra 法：ラベル

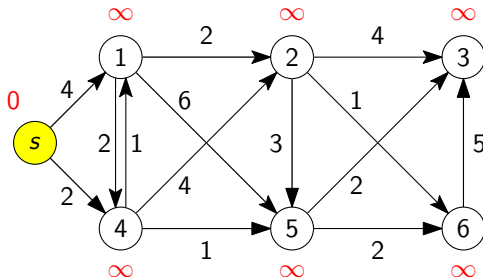
白：未確定頂点，黄：確定頂点



初期化：頂点 s のラベルは 0，他の頂点のラベルはどれも ∞

実際の Dijkstra 法：未確定頂点の選択

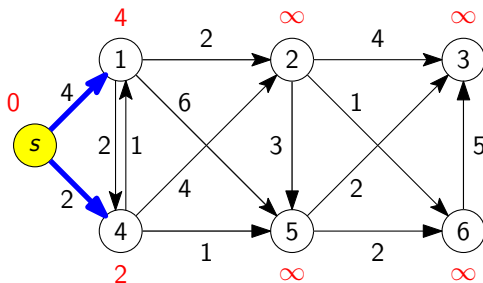
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新

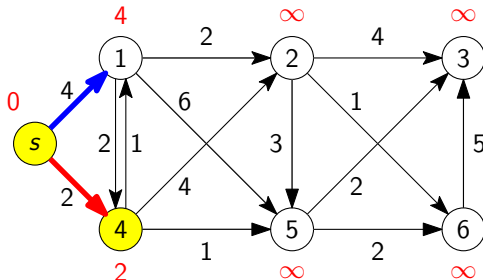
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で，最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (2)

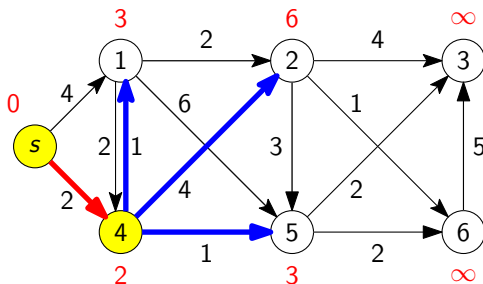
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (2)

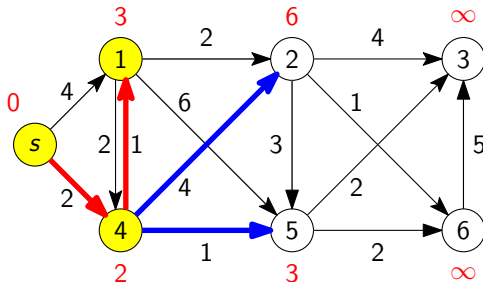
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (3)

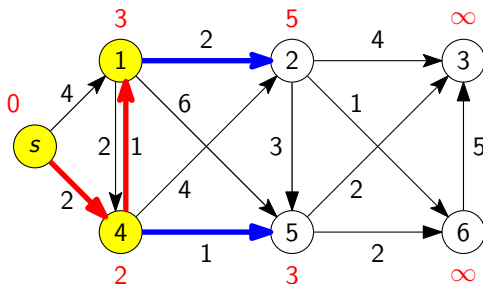
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (3)

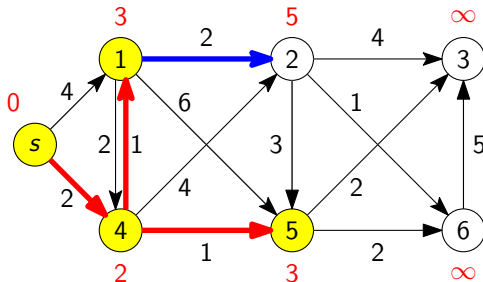
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (4)

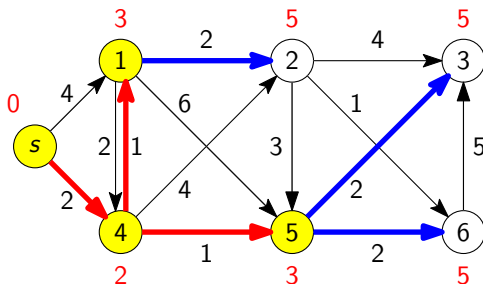
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (4)

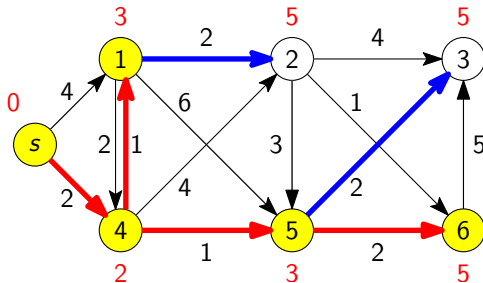
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (5)

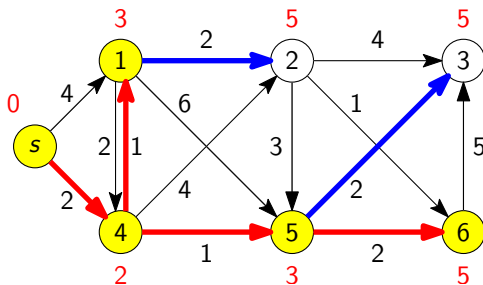
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (5)

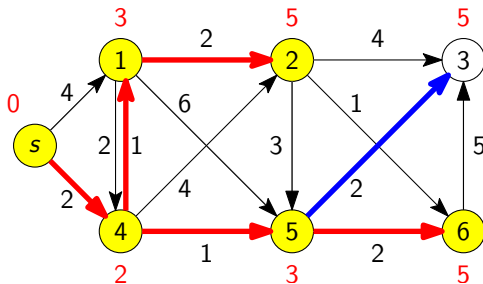
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (6)

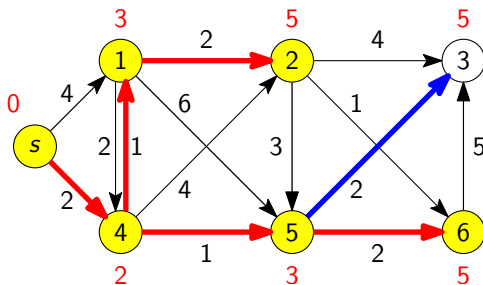
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (6)

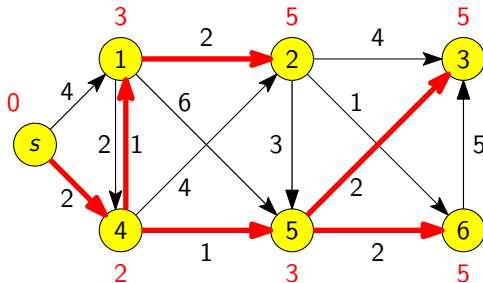
白：未確定頂点，黄：確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：未確定頂点の選択 (7)

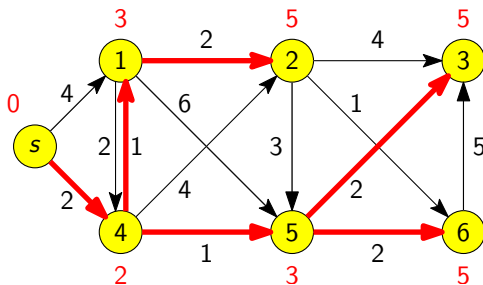
白：未確定頂点，黄：確定頂点



未確定頂点の中でラベル最小のものを選び，確定頂点にする

実際の Dijkstra 法：ラベルの更新 (7)

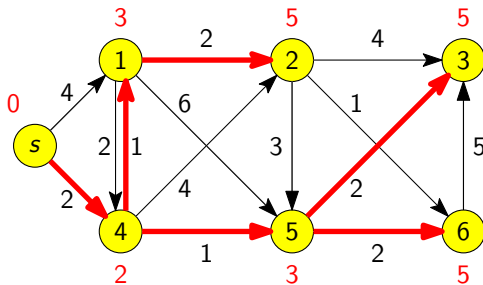
白： 未確定頂点，黄： 確定頂点



選ばれた頂点を始点とする辺の終点である頂点のラベルを更新する
(確定頂点を通る道の中で、最短のものの長さがラベルとなる)

実際の Dijkstra 法：アルゴリズムの停止

白：未確定頂点，黄：確定頂点



すべての頂点が確定頂点となったので，停止する

- このときに，最短路木が得られる

目次

- 1 最短路問題とは？
- 2 最短路の重要な性質
- 3 辺長がすべて同じ場合：幅優先探索
- 4 一般の場合：Dijkstra 法
- 5 今日のまとめと今後の予告

今日のまとめと今後の予告

今日のまとめ：今日の目標

- 最短路問題の定義と解法を理解する
- 最短路問題を解けるようになる

今後の予告：ネットワークに関わる 3 つの最適化問題

済 最短路問題

- 最大流問題
- 最小費用流問題

注：ネットワークに関わる最適化問題は他にもたくさんある

目次

- ① 最短路問題とは？
- ② 最短路の重要な性質
- ③ 辺長がすべて同じ場合：幅優先探索
- ④ 一般の場合：Dijkstra 法
- ⑤ 今日のまとめと今後の予告