アルゴリズムI

中野

Note 5 グラフアルゴリズム

作成　2020.5.28

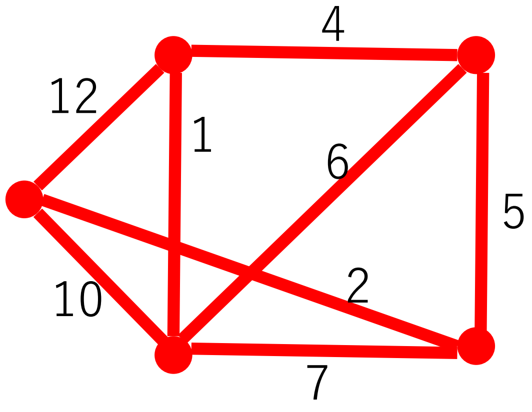
Update 2020.6.4　6.25 7.1 7.3 7.9　2021.6.8 6.16

最短路問題

グラフG=(V,E)と、始点s∈Vが与えられたとする。
Gの各辺(u,v)∈Eの長さを、w(u,v)≧0とする。
このとき、始点sから、
Gの各点までの最短の距離を求めたい。

最短路問題

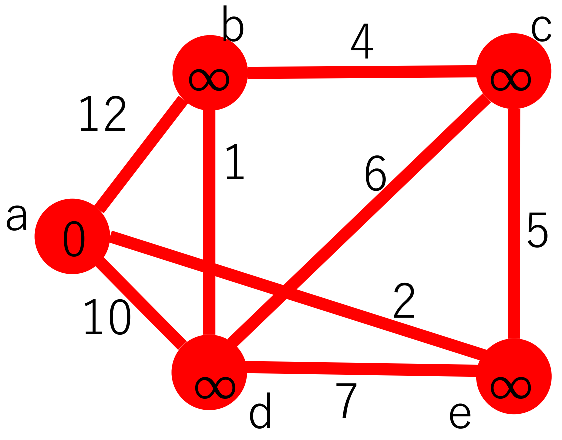
グラフG=(V,E)と、始点s∈Vが与えられたとする。
Gの各辺(u,v)∈Eの長さを、w(u,v)≧0とする。
このとき、始点sから、
Gの各点までの最短の距離を求めたい。

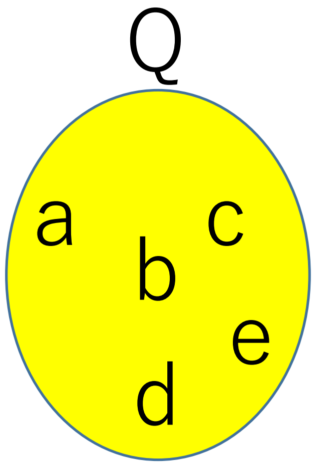


ダイキストラ法で解きましょう！
イメージ　波紋が広がるように。。。

最短路問題　ダイキストラ法　具体例

```
Dijkstra(G,w,s)
{初期化}
for 各点 u ∈ V[G] do
        /*  V{G] はグラフGの点集合  */
      d[u] ← ∞
      π[u] ← NIL
  S = φ
  Q = V
  d[s] ← 0
```





ダイキストラ法　概要

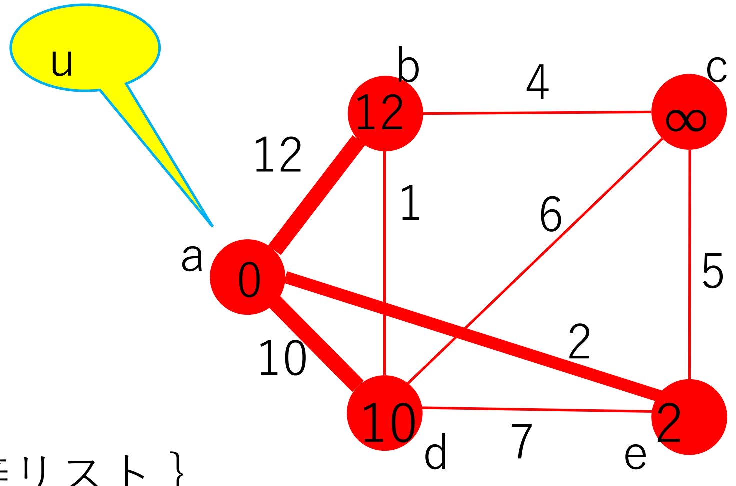
既に始点からの最短距離が決定した点の集合をSとする。最短距離が未定の点の集合をQとする。
最初は、S=φ であり、これに最短距離が決定した点を1点づつ順次追加していく。
最後にS = V となり終了する。

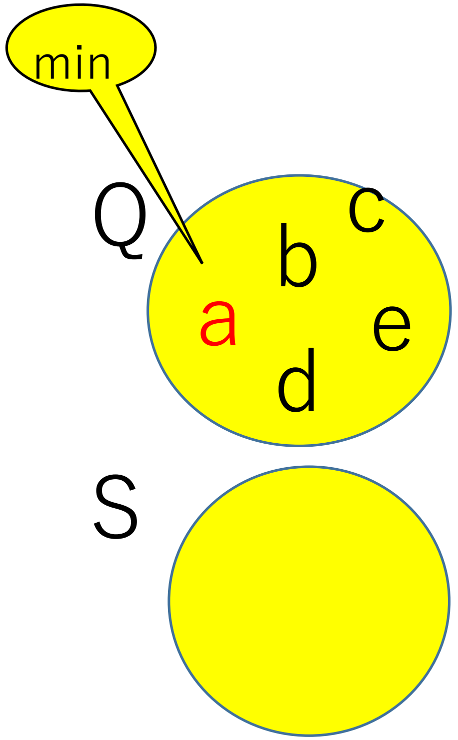
最短路問題　ダイキストラ法　具体例

Dijkstra 後半

{各点までの最短距離を計算する}

```
    while Q ≠ φ do
do  u ← Extract-Min(Q)
        {d(u)が最小の点uを,集合Qから取り出す}
        S ← S ∪ { u }
        for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
              if d[v] ＞ d[u] + w(u,v) then
                    d[v] ← d[u] + w(u,v)
                    π[v] ← u              {始点からvへの道でvの直前の点はu}
```





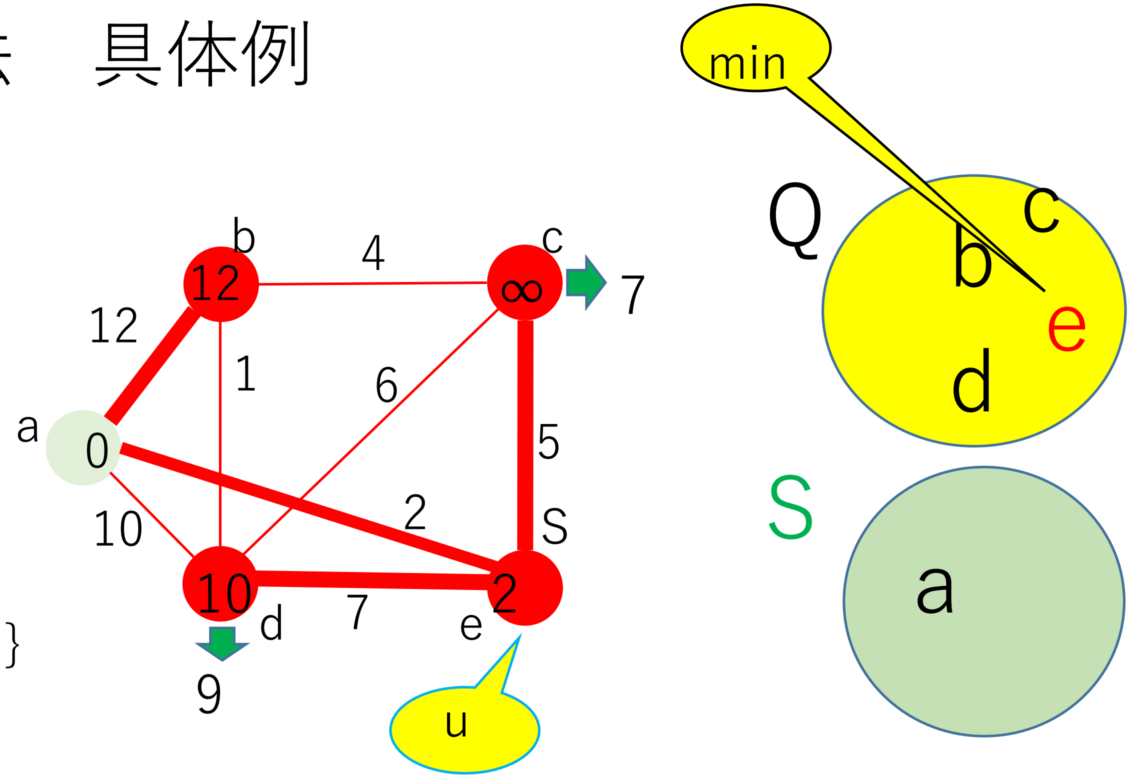
ダイキストラ法　概要

Sに点uが含まれるときは、始点sから点uまでの最短距離を d(u)とする。
Sに点uが含まれないときは、S中の点のみを中継点として許すような、
始点sから点uまでの、(暫定)最短距離をd(u)とする。
始点sからvへの最短路のvの直前の点は π[v]=uとする。

最短路問題　ダイキストラ法　具体例

Dijkstra 後半

```
{各点までの最短距離を計算する}
   while Q ≠ ϕ do
do   u ← Extract-Min(Q)
        {d(u)が最小の点uを,集合Qから取り出す}
         S ← S ∪ { u }
         for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
               if d[v] ＞ d[u] + w(u,v) then
                    d[v] ← d[u] + w(u,v)
                    π [v] ← u
```

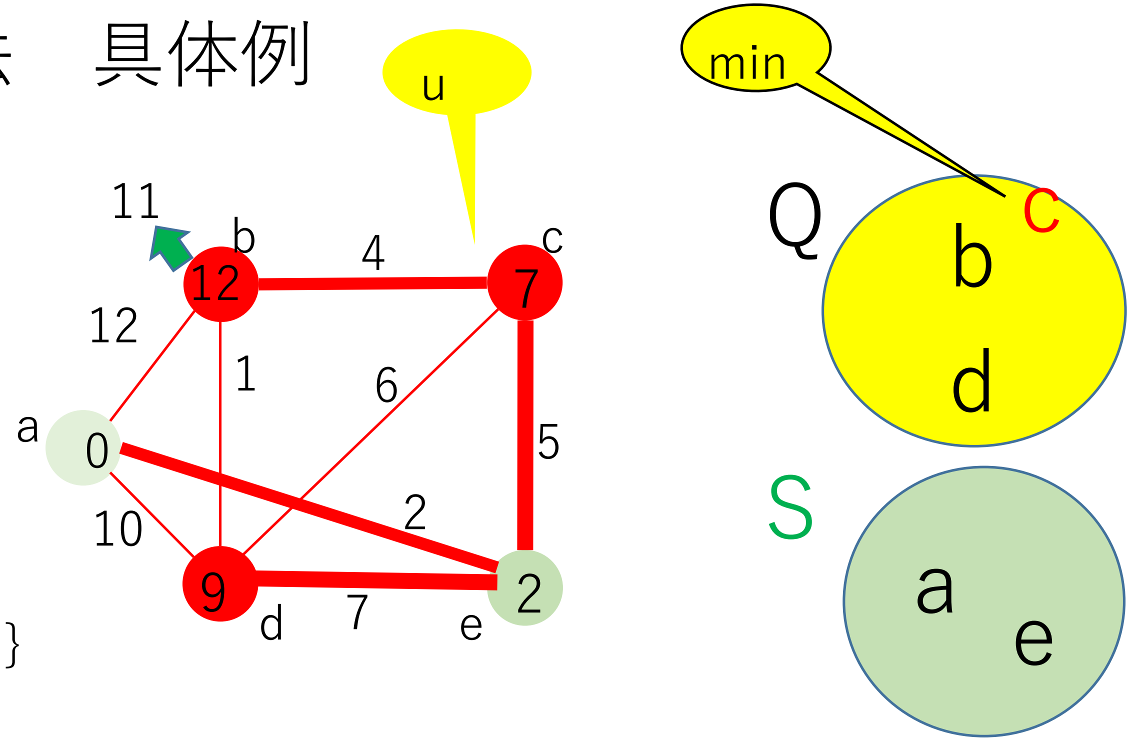


最短路問題

ダイキストラ法　具体例

Dijkstra 後半

```
{各点までの最短距離を計算する}
   while Q ≠ ϕ do
do   u ← Extract-Min(Q)
       {d(u)が最小の点uを,集合Qから取り出す}
        S ← S ∪ { u }
        for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
                if d[v] ＞ d[u] + w(u,v) then
                        d[v] ← d[u] + w(u,v)
                        π [v] ← u
```



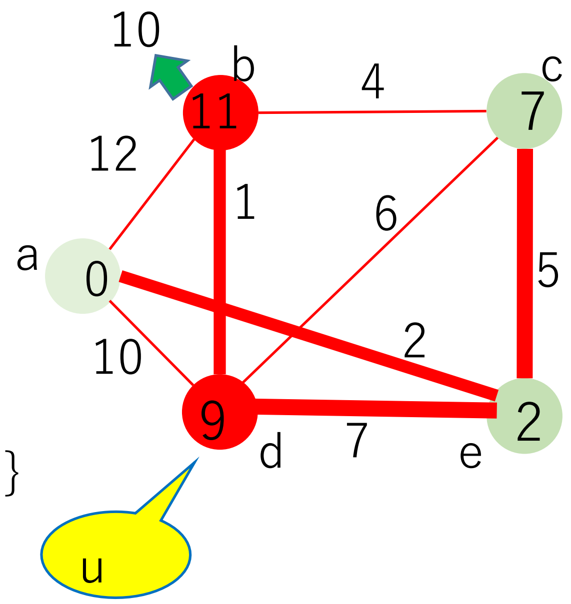
最短路問題

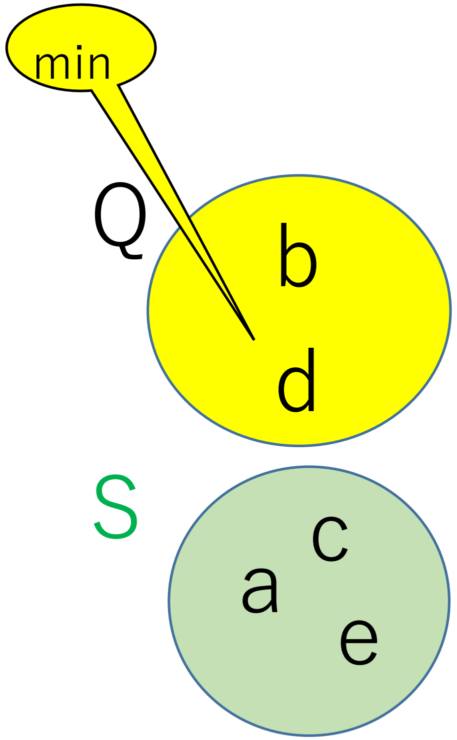
ダイキストラ法　具体例

Dijkstra 後半

{各点までの最短距離を計算する}

```
    while Q ≠ φ do
do   u ← Extract-Min(Q)
       {d(u)が最小の点uを,集合Qから取り出す}
        S ← S ∪ { u }
        for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
                if d[v] > d[u] + w(u,v) then
                        d[v] ← d[u] + w(u,v)
                        π[v] ← u
```



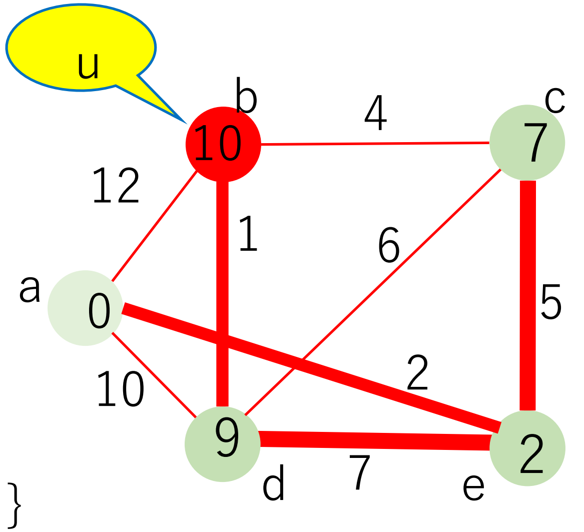


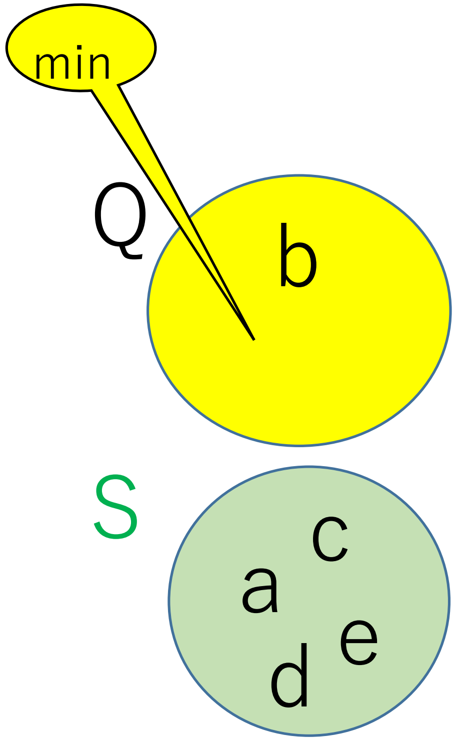
最短路問題

ダイキストラ法　具体例

Dijkstra 後半
{各点までの最短距離を計算する}

```
while Q ≠ ϕ do
do   u ← Extract-Min(Q)
     {d(u)が最小の点uを,集合Qから取り出す}
      S ← S ∪ { u }
      for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
             if d[v] > d[u] + w(u,v) then
                    d[v] ← d[u] + w(u,v)
                    π [v] ← u
```





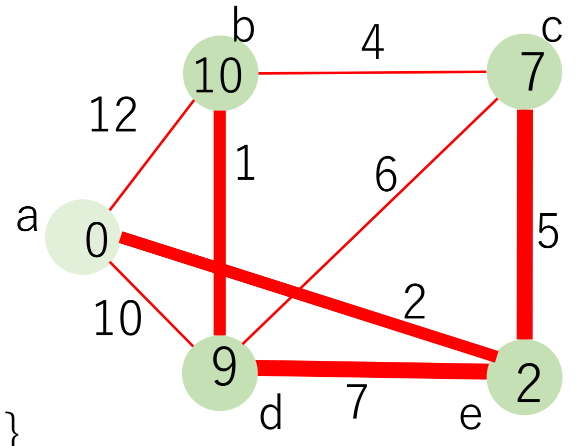
最短路問題

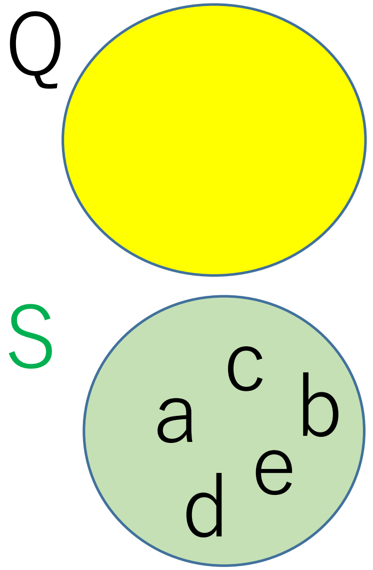
ダイキストラ法　具体例　おしまい

Dijkstra 後半

{各点までの最短距離を計算する}

```
while Q ≠ ϕ do
do   u ← Extract-Min(Q)
       {d(u)が最小の点uを,集合Qから取り出す}
        S ← S ∪ { u }
        for 各点 v ∈ Adj[u] do { Adj[u]は点uの隣接リスト }
              if d[v] > d[u] + w(u,v) then
                   d[v] ← d[u] + w(u,v)
                   π[v] ← u
```





正当性

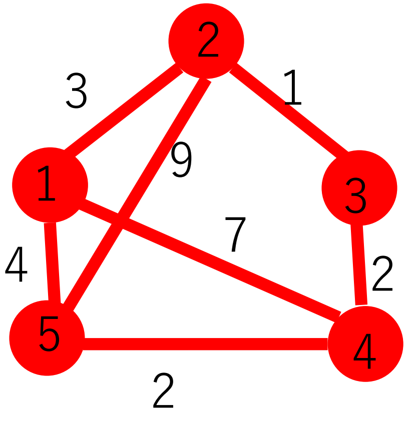
計算時間

ヒープ利用時の計算時間　O((V + E) log V)

全点間最短路問題 その1

グラフ G=(V,E)と、Gの各辺$(u,v) \in E$の長さ $w(u,v) \geqq 0$ が与えられたとする。すべての2点間の(最短の)距離を求めよう。

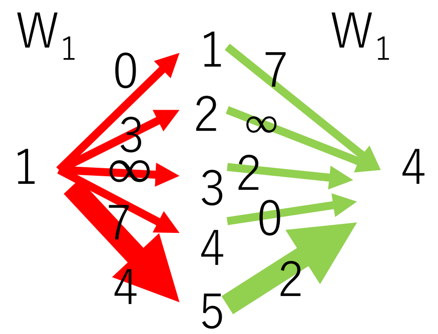
w(u,v)から構成される2次元配列をW_1とする。入力である。



W_1

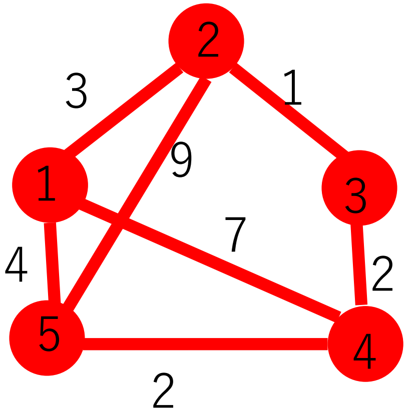
	1	2	3	4	5
1	0	3	∞	7	4
2	3	0	1	∞	9
3	∞	1	0	2	∞
4	7	∞	2	0	2
5	4	9	∞	2	0

全点間最短路問題 その1



w(u,v)から構成される2次元配列をW_1とする。入力である。

高々2辺の経路の最短距離の表W_2を計算しよう。



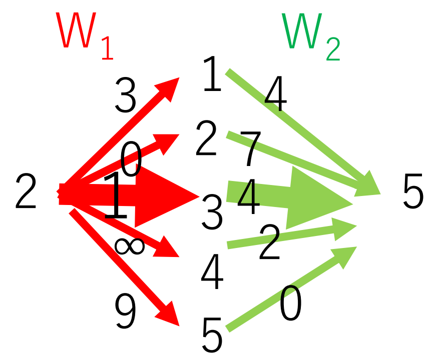
W_1

	1	2	3	4	5
1	0	3	∞	7	4
2	3	0	1	∞	9
3	∞	1	0	2	∞
4	7	∞	2	0	2
5	4	9	∞	2	0

W_2

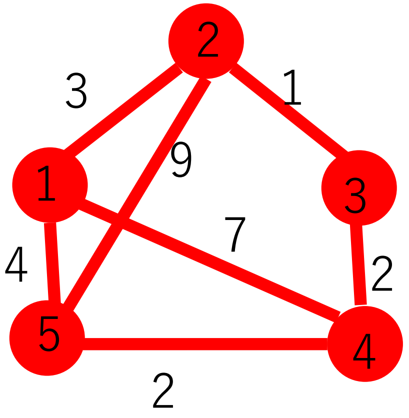
	1	2	3	4	5
1	0	3	4	6	4
2	3	0	1	3	7
3	4	1	0	2	4
4	6	3	2	0	2
5	4	7	4	2	0

全点間最短路問題 その1



w(u,v)から構成される2次元配列をW_1とする。入力である。

高々3辺の経路の最短距離の表W_3を計算しよう。

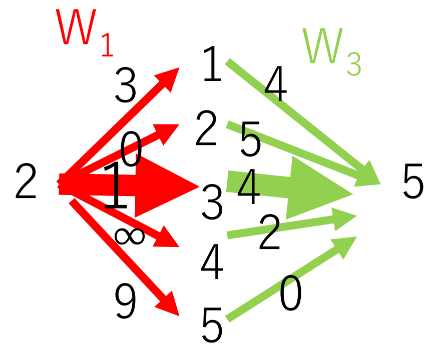


W_1	1	2	3	4	5
1	0	3	∞	7	4
2	3	0	1	∞	9
3	∞	1	0	2	∞
4	7	∞	2	0	2
5	4	9	∞	2	0

W_2	1	2	3	4	5
1	0	3	4	6	4
2	3	0	1	3	7
3	4	1	0	2	4
4	6	3	2	0	2
5	4	7	4	2	0

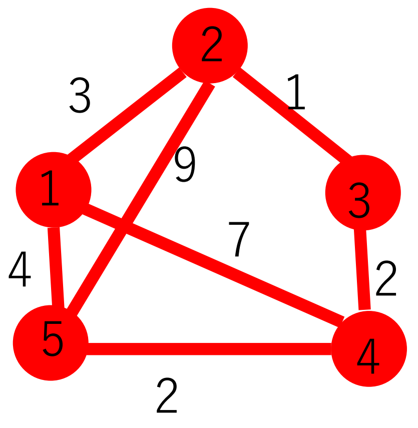
W_3	1	2	3	4	5
1	0	3	4	6	4
2	3	0	1	3	5
3	4	1	0	2	4
4	6	3	2	0	2
5	4	5	4	2	0

全点間最短路問題 その1



w(u,v)から構成される2次元配列をW_1とする。入力である。

高々4辺の経路の最短距離の表W_4を計算しよう。



W_1

	1	2	3	4	5
1	0	3	∞	7	4
2	3	0	1	∞	9
3	∞	1	0	2	∞
4	7	∞	2	0	2
5	4	9	∞	2	0

W_3

	1	2	3	4	5
1	0	3	4	6	4
2	3	0	1	3	5
3	4	1	0	2	4
4	6	3	2	0	2
5	4	5	4	2	0

W_4

略

$W_5 = W_4$

どの最短経路も辺は高々4本

(n-1)本

全点間最短路問題 その1

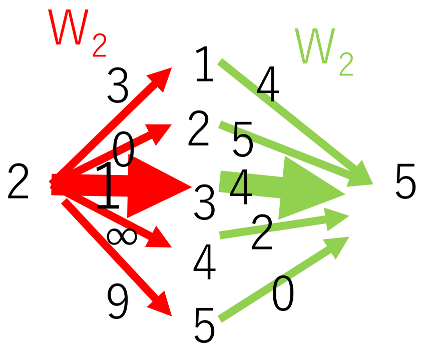
Shortest-Paths-1(W)

n ← rows[W] { Wの列の個数を nとする }
for m ← 2 to n-1 do {W_m を求める。W_1 = Wである。}
    for i ← 1 to n do
        for j ← 1 to n do
            $e^{m}_{ij} \leftarrow e^{m-1}_{ij}$ { W_mのi行j列要素e^{m}_{ij}の初期化 }
            for k ← 1 to n do
                $e^{m}_{ij} \leftarrow \min\{ e^{m}_{ij}, e^{1}_{ik} + e^{m-1}_{kj} \}$

正当性 DPである。
計算時間 $O(n^4)$ 表を1枚計算するのに $O(n^3)$時間 表はn-1枚作製する

全点間最短路問題 その2

W_4を計算する
W_4 :高々4本のパスの最短距離の表



W_3は不要

W_1 => W_2 => W_3 => W_4 => W_5 => …=>W_{n-1}

W_1 => W_2 =>W_4 => W_8 => W_{16}=>…=>W_k

(kはn-1以上の最小の２のべき乗

計 log n枚の表)

正当性

計算時間 O (n³ log n)　　表を1枚計算するのに　O(n³)時間　表はlog n枚作製する

全点間最短路問題 その2

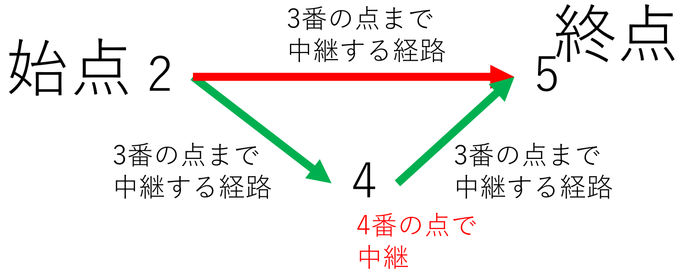
Shortest-Paths-1(W_1)

$n \leftarrow$ rows[W] { Wの列の個数を nとする }
m =1
while m ≦ n-1 do {W_m を求める, m=1,2,4,8,…}
    for i ← 1 to n do {始点}
        for j ← 1 to n do {終点}
            $e^m_{ij} \leftarrow e^{m-1}_{ij}$ { W_mのi行j列の要素e^m_{ij}の初期化 }
            for k ← 1 to n do
                $e^m_{ij} \leftarrow \min\{ e^m_{ij}, e^{m/2}_{ik} + e^{m/2}_{kj} \}$
    m=2m

正当性 DPである。
計算時間 $O(n^3 \log n)$ 表を1枚計算するのに $O(n^3)$時間 表はlog n枚作製する

全点間最短路問題 その3

D_4 :4番の点まで中継点としてよ
最短距離



Shortest-Paths-1(W)

n ← rows[W] ｛Wの列の個数を nとする｝

D_0 = W {D_0: 0番の点までを中継点に使用してよい最短距離の表}

for k ← 1 to n do {k番の点までを中継点に使用してよい)

{D_k を求める。D_k: K番の点までを中継に使用してよい最短距離の表}

for i ← 1 to n do {始点}

for j ← 1 to n do {終点}

$e^k_{ij} \leftarrow \min\{\ e^{k-1}_{ij},\ \ e^{k-1}_{ik} + e^{k-1}_{kj}\ \}$

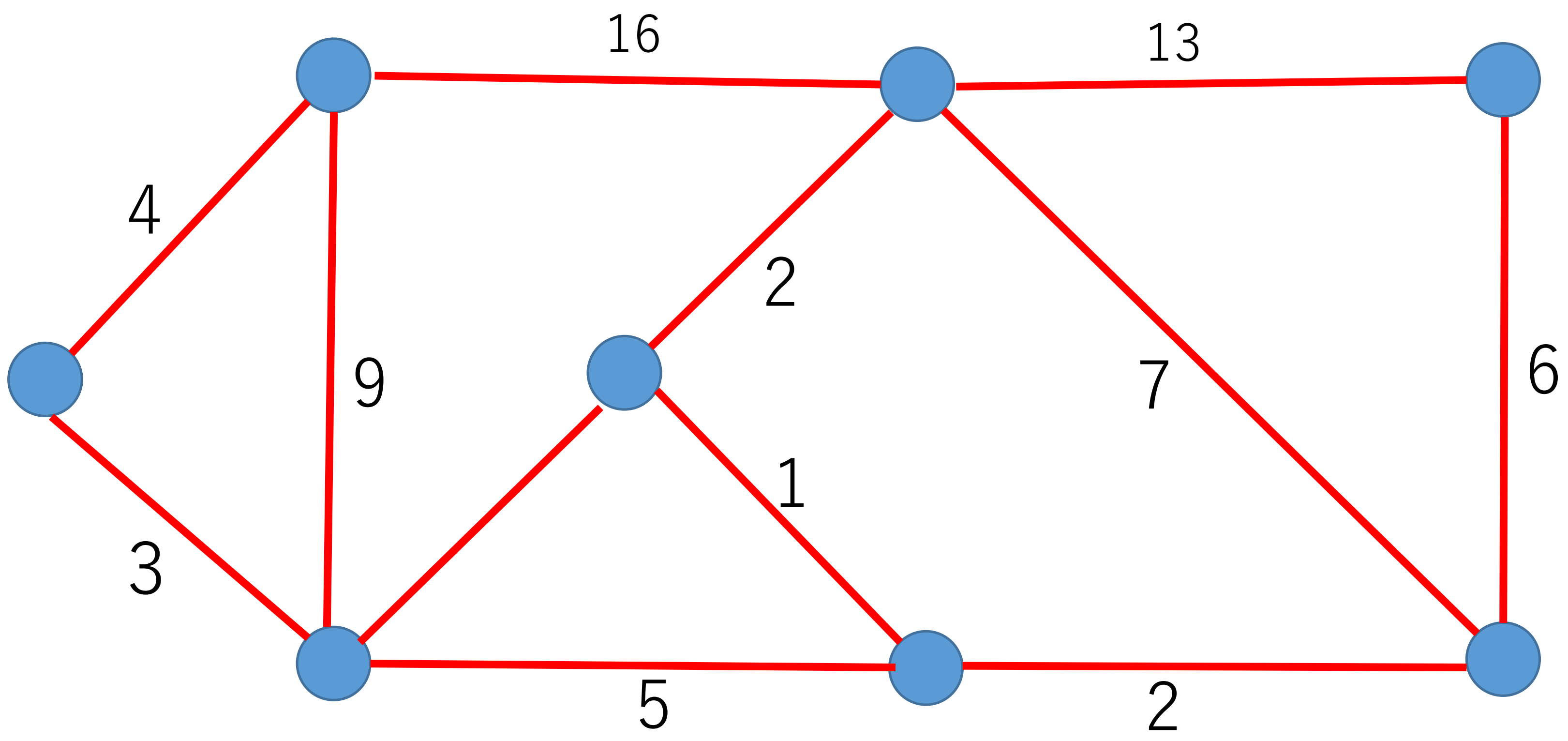
{ k番の点を使わない or k 番の点を使う のどちらか}

正当性

計算時間 O (n^3)　　表を1枚計算するのに　O(n^2)時間　表はn枚作製する

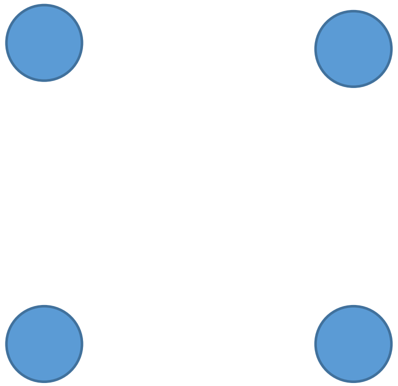
MST　(最小全域木　Minimum Spanning Tree)

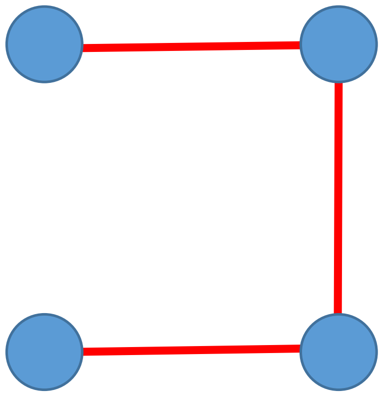
全点をむすぶ辺の重さの合計が最小の木（最小全域木）を計算したい
(全部の点を最短の高速道路で結ぶには



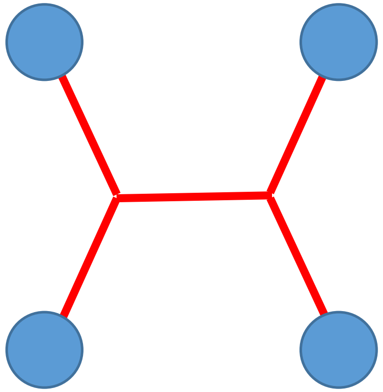
最小全域木、スタイナー木

4つの町を結ぶ最短の高速道路は？





最小全域木



スタイナー木

MST　(最小全域木　Minimum Spanning Tree)

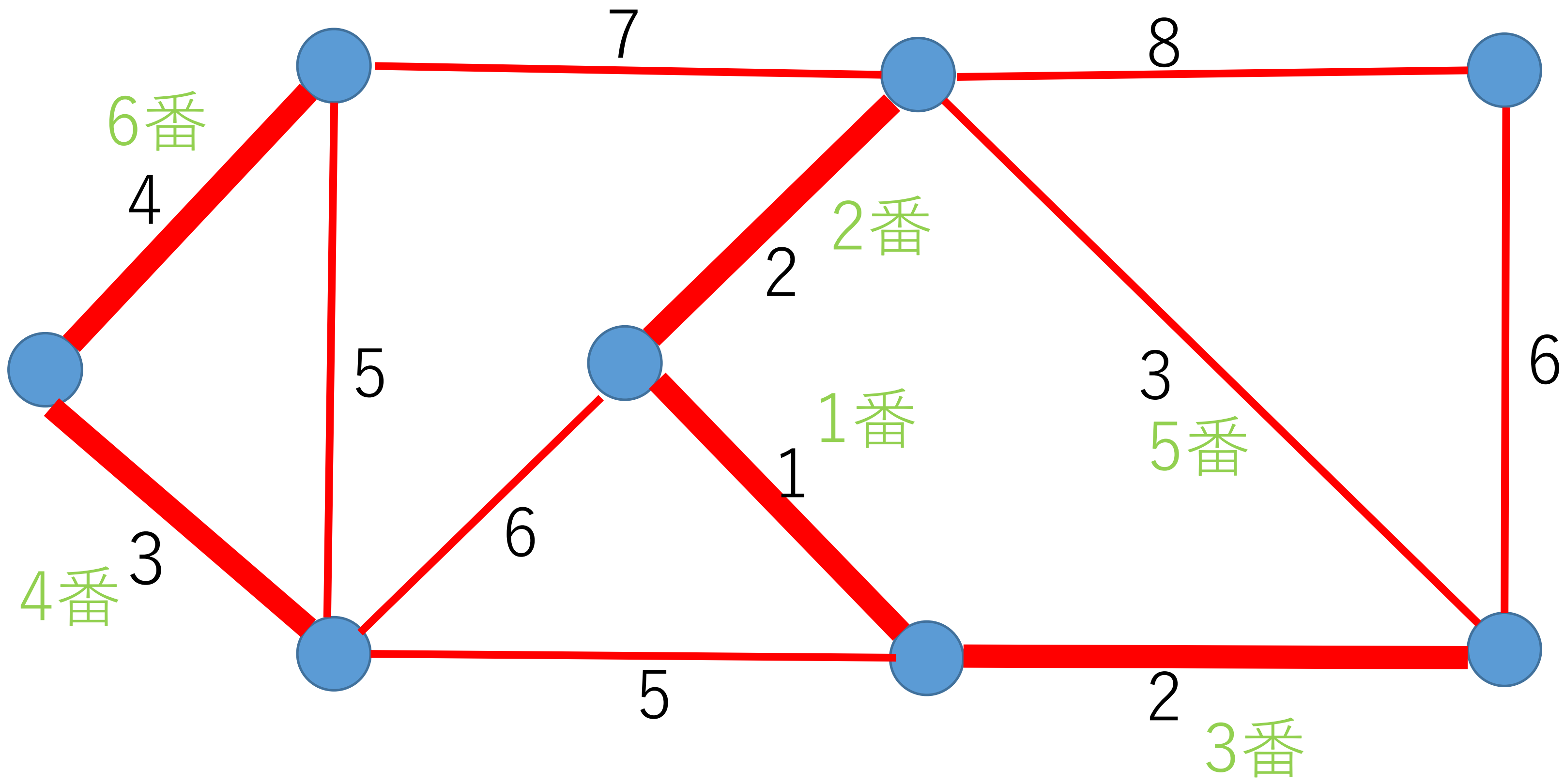
```
MST-Kruskal(G=(V,E),w)
    A ← φ { 初期化 }
    for 各点 v ∈ V do
        Make-Set(v)  { 各 1 点v からなる集合を作る }{各集合の要素数は1}
    E中の辺を、wの小さな順にソートする
    for 各辺 (u,v) ∈ E do {上のソート順に実行する}
      if 点u と点vが異なる点集合に属しているならば then
          A ← A ∪ { (u,v) }  {解にこの辺を追加する}
          点uの属している集合と、点vの属している集合とを合体して
          新しい集合を作る(集合内の最小の点の番号を集合の番号とする)
    return A
```

$O(E \log E + V^2)$

辺のソートで E log E
集合の合体で　V^2

クラスカルのアルゴリズム（具体例）

全部の点を最短の高速道路で結ぶには



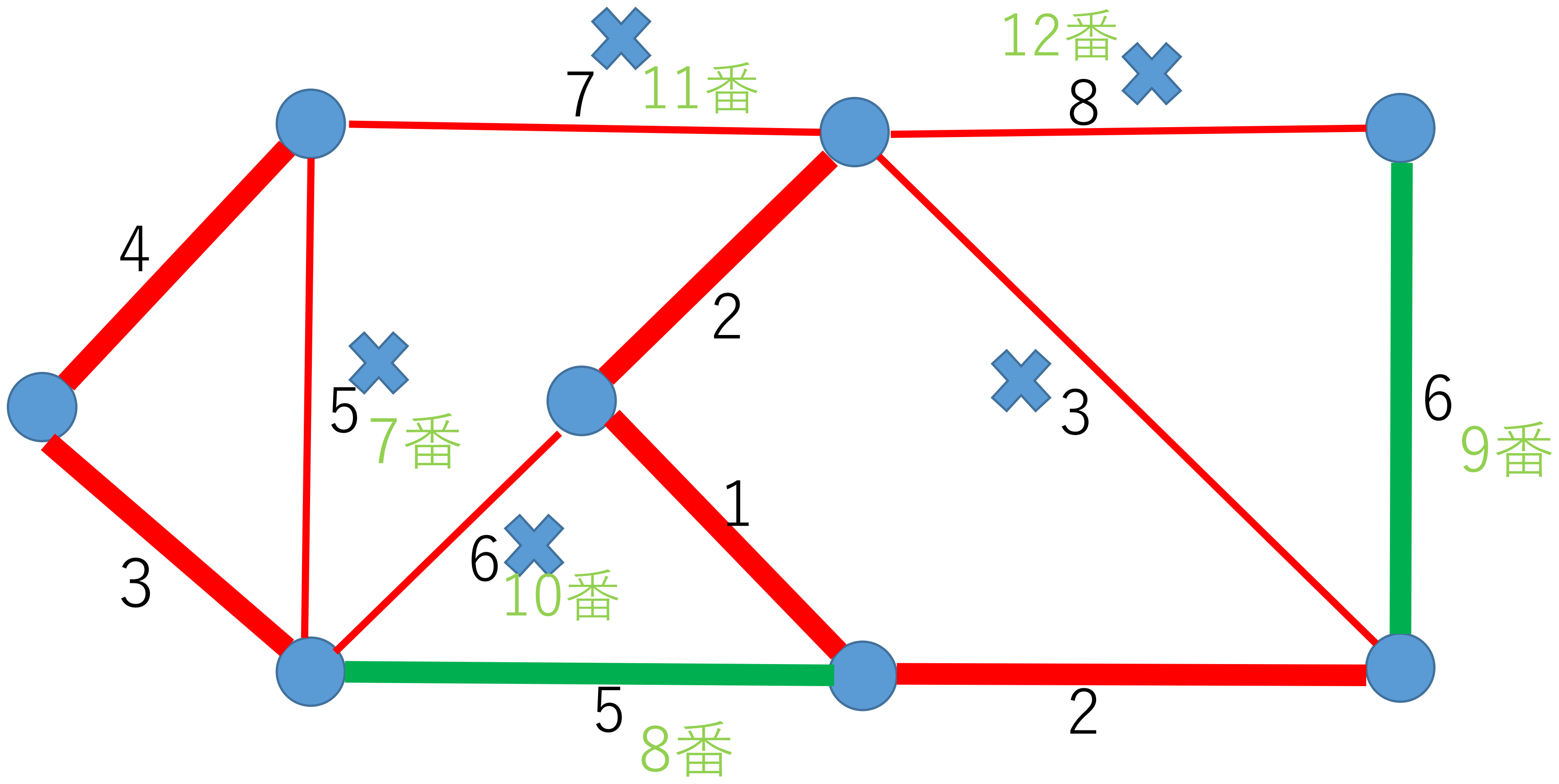
辺の長さの
小さい順に
チェック

異なる島を
結ぶなら解
に追加する

そうでなけ
れば捨てる

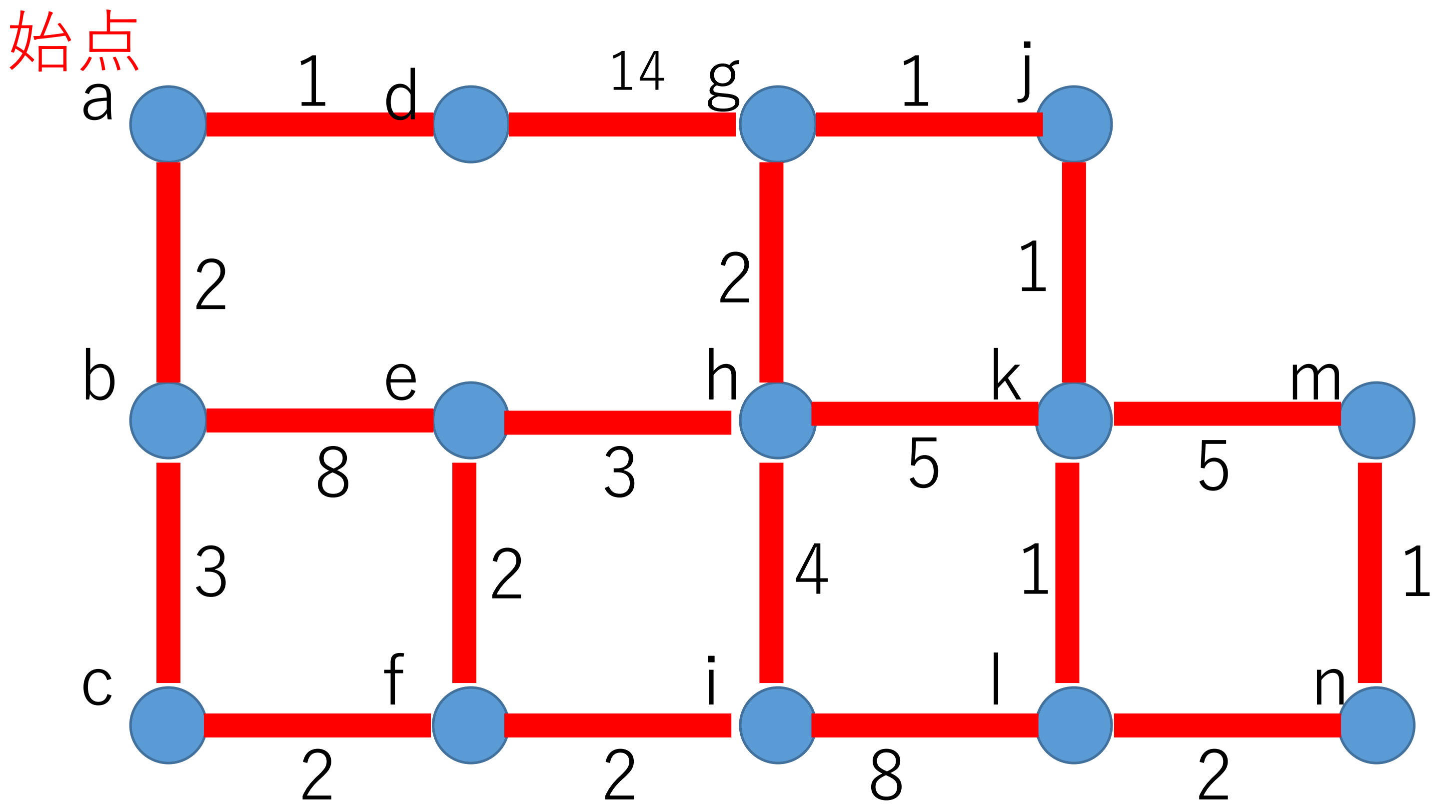
クラスカルのアルゴリズム（具体例　つづき）

全部の点を最短の高速道路で結ぶには



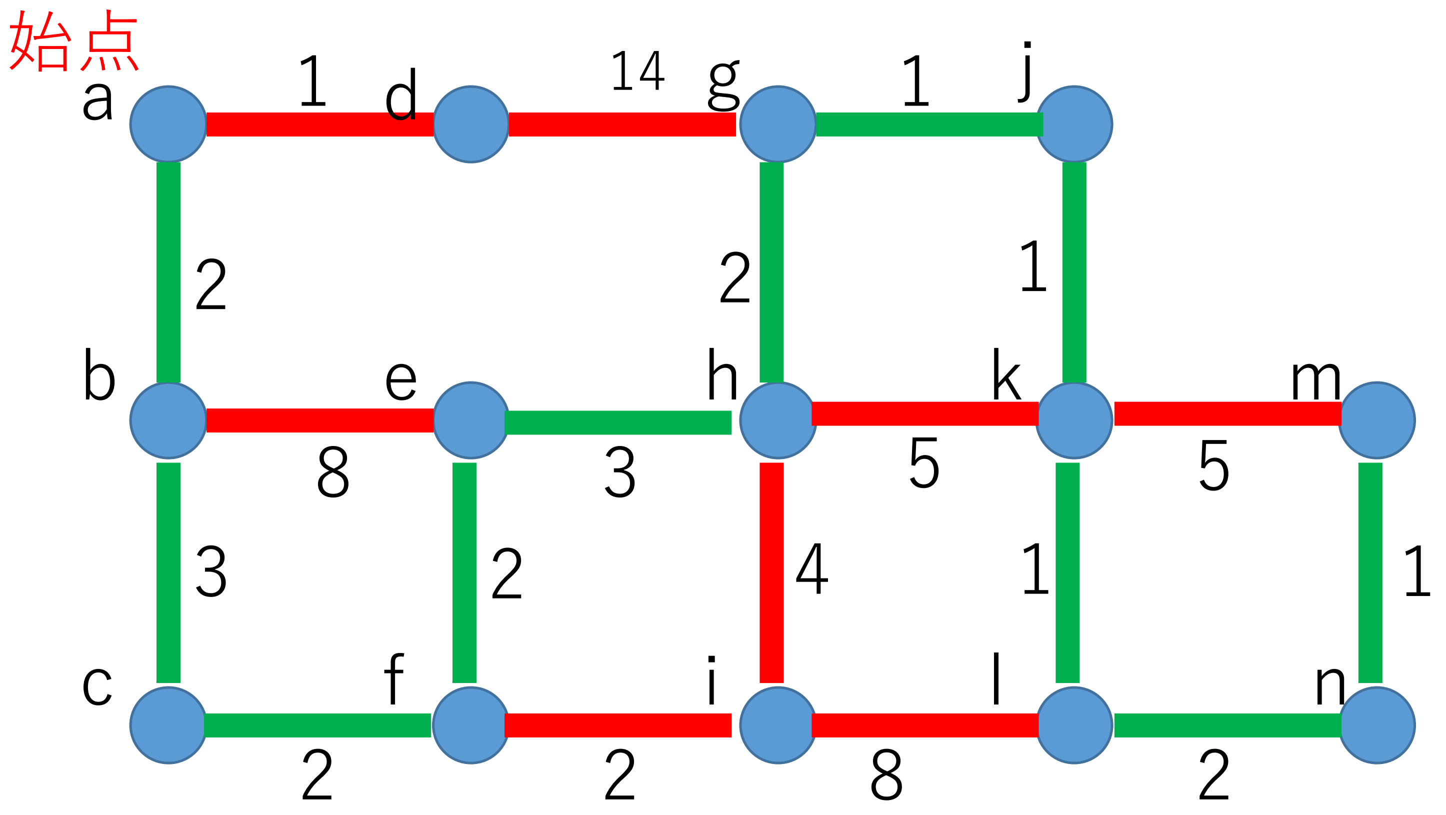
理解確認クイズ

(1)点aを始点とする。点mまでの最短路の長さを求めよ。



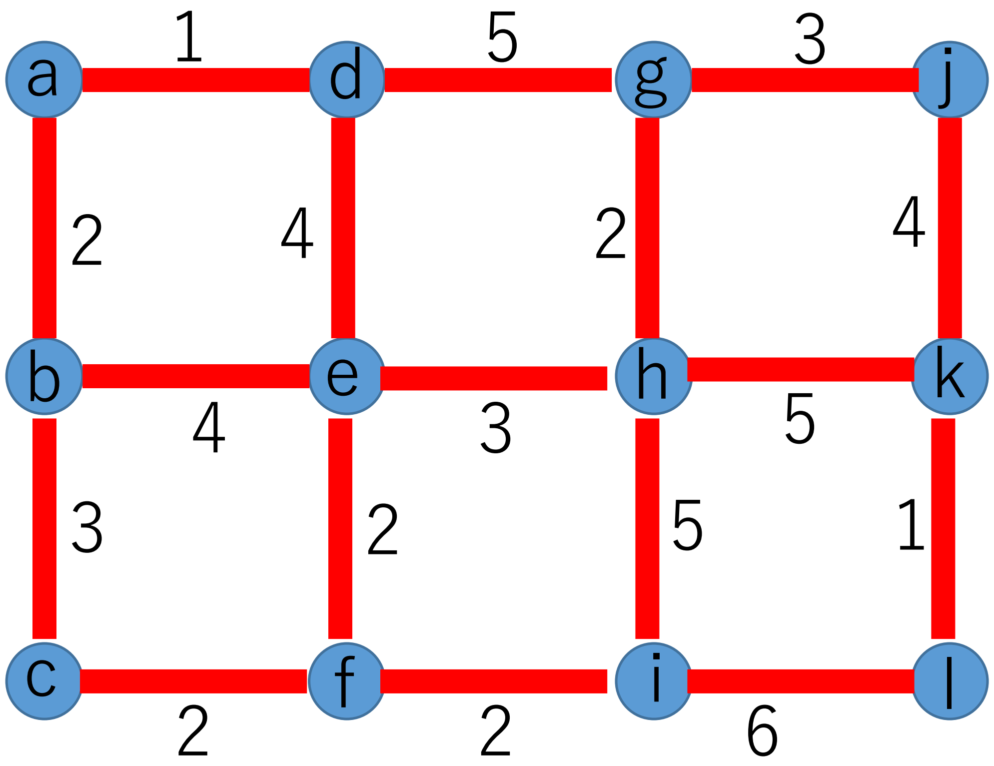
理解確認クイズ

(1)点aを始点とする。点mまでの最短路の長さを求めよ。 ２０



理解確認クイズ

(2)最小全域木を求めよ。その重さはいくらか？



理解確認クイズ

(2)最小全域木を求めよ。その重さはいくらか？　　25