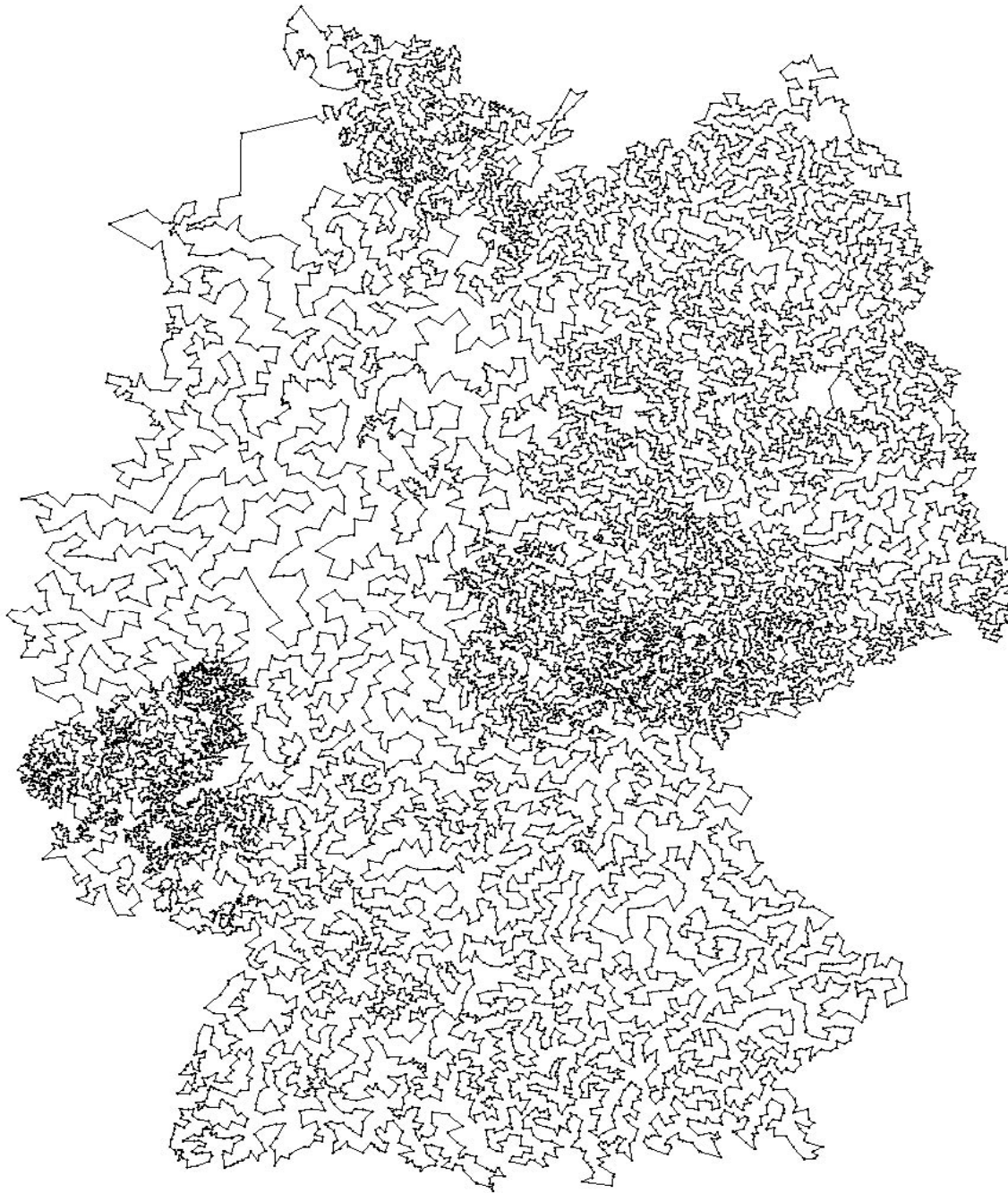
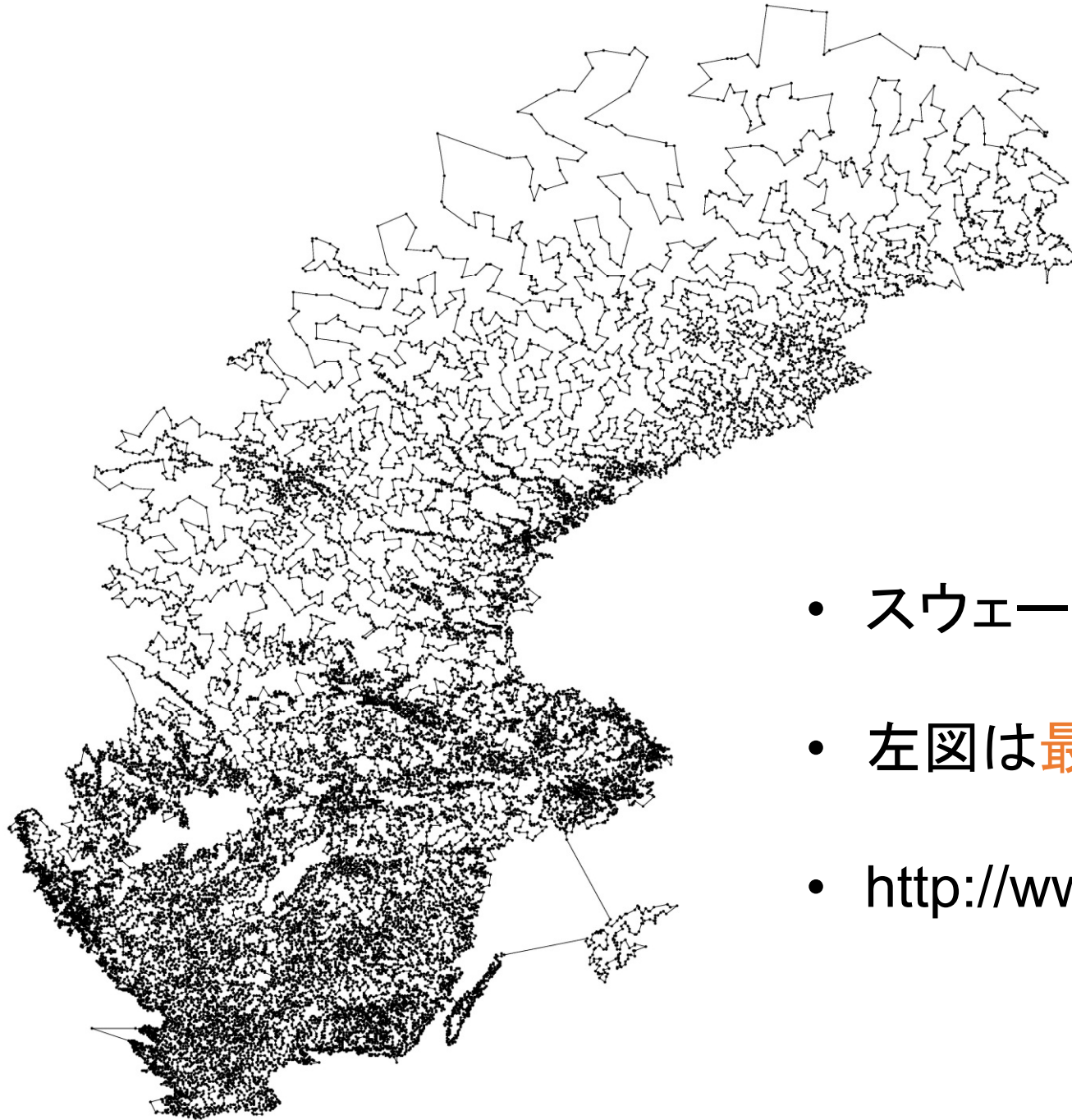


巡回セールスマン問題



- 各目的地を一度ずつ訪問
- 総移動距離を最小化
- ドイツ15112都市の例
- 左図は最適解
- <http://www.tsp.gatech.edu>

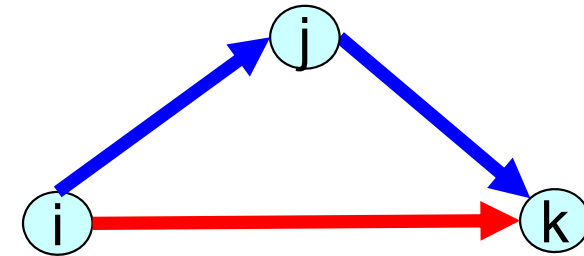
巡回セールスマン問題



- スウェーデン24978都市の例
- 左図は最適解
- <http://www.tsp.gatech.edu/>

距離に関する仮定

- 距離は非負: $d(i,j) \geq 0$ ($\forall i, j$)
- 対称: $d(i,j) = d(j,i)$ ($\forall i, j$)
- 三角不等式を満たす: $d(i,k) \leq d(i,j) + d(j,k)$ ($\forall i, j, k$)



仮定を満たす距離の例:

- ユークリッド距離 (ℓ_2 距離): $\sqrt{\sum_i (x_i - y_i)^2}$
- マンハッタン距離 (ℓ_1 距離): $\sum_i |x_i - y_i|$
- チェビシェフ距離 (ℓ_0 距離): $\max_i |x_i - y_i|$
- 文字列に対するハミング距離: 2つの文字列の距離 = 異なる文字数
 $d(\text{make}, \text{cakes}) = 2, \quad d(\text{shimoda}, \text{shioura}) = 3$

など多数

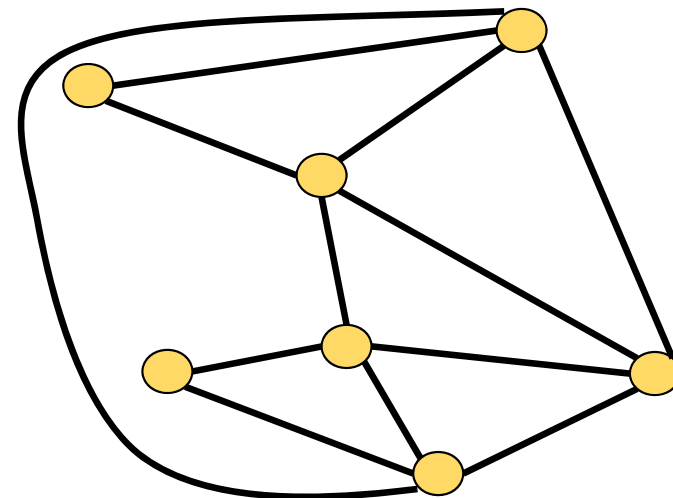
2種類の近似アルゴリズム

- 最小全域木を用いた2近似アルゴリズム
- 最小全域木＋最小重みマッチングを用いた1.5近似アルゴリズム

準備：アルゴリズムでは以下の性質を利用

定理 連結な無向グラフにおいて、各頂点の次数が偶数ならば、全ての枝を一筆書きでたどることができる(**オイラー閉路**とよぶ)

(雑な証明) 上手に枝をたどって
いけば、必ず一筆書きできる
(なぜ?)



最小全域木を用いた2近似アルゴリズム

最小全域木を用いた2近似アルゴリズム

ステップ1: 最小全域木を求める

ただし, 次のグラフを考える.

V =全ての点集合

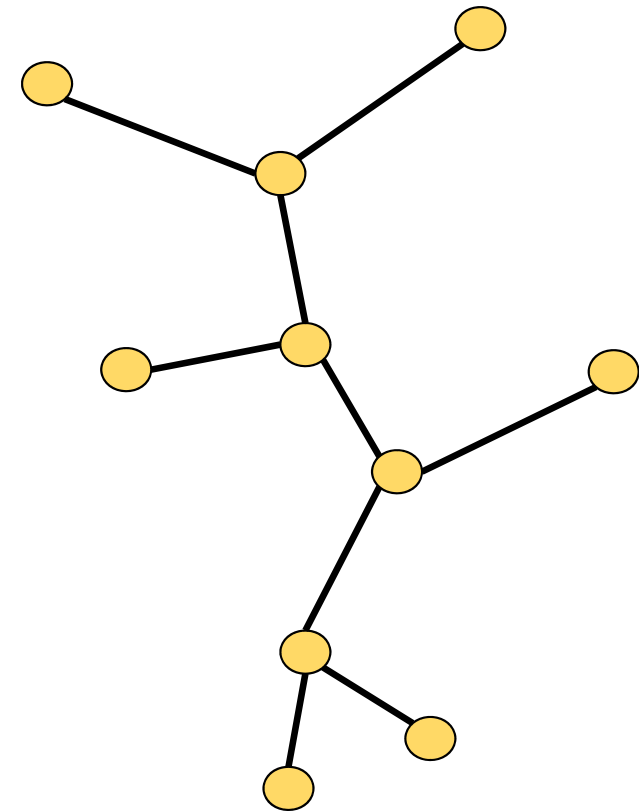
E =全ての異なる点の対

枝 (i,j) の費用 $=d(i,j)$

ステップ2: 最小全域木の枝を全て二重にする.

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところをショートカットして, 巡回路にする.



最小全域木を用いた2近似アルゴリズム

ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V =全ての点集合

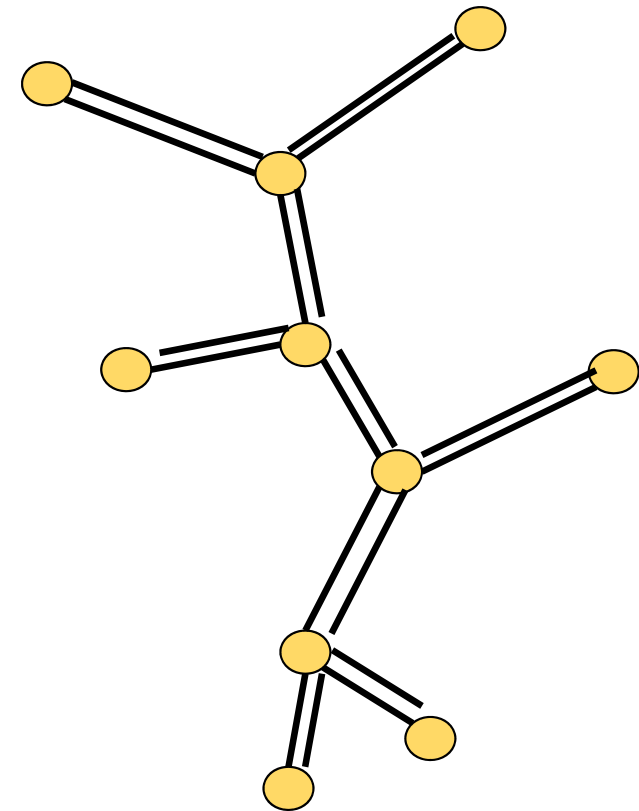
E =全ての異なる点の対

枝 (i,j) の費用 $=d(i,j)$

ステップ2: 最小全域木の枝を全て二重にする.

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



最小全域木を用いた2近似アルゴリズム

ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V =全ての点集合

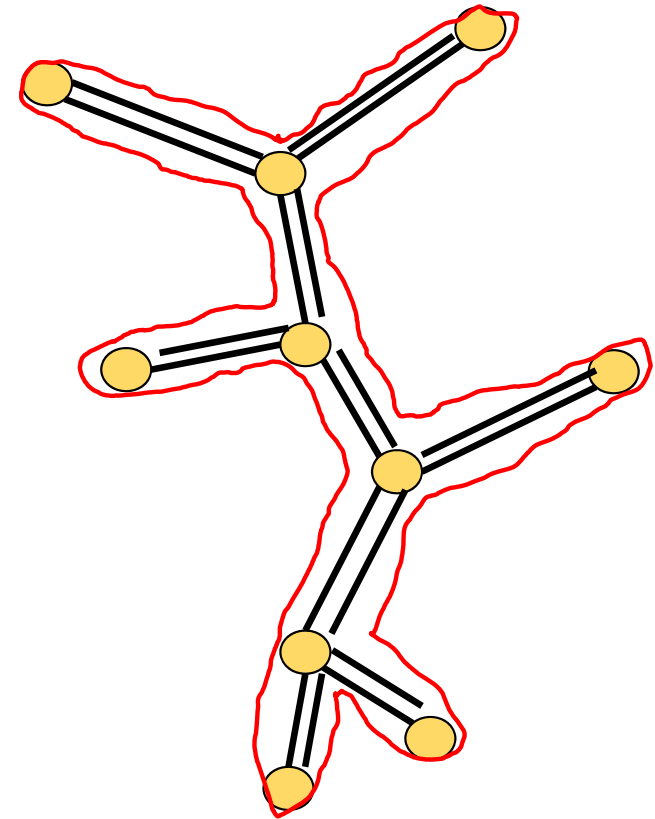
E =全ての異なる点の対

枝 (i,j) の費用= $d(i,j)$

ステップ2: 最小全域木の枝を全て二重にする.

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



最小全域木を用いた2近似アルゴリズム

ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V =全ての点集合

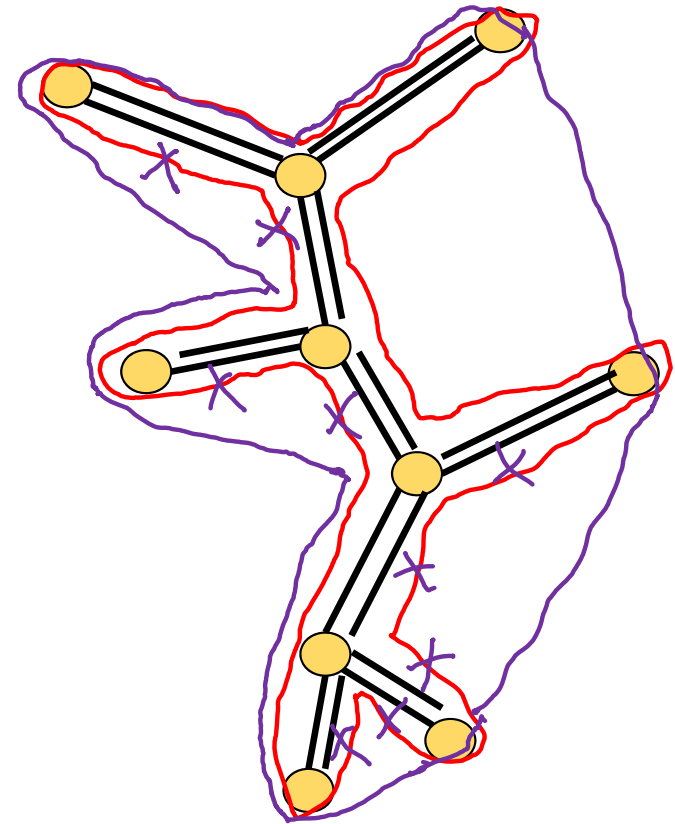
E =全ての異なる点の対

枝 (i,j) の費用= $d(i,j)$

ステップ2: 最小全域木の枝を全て二重にする.

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



アルゴリズムの近似比解析

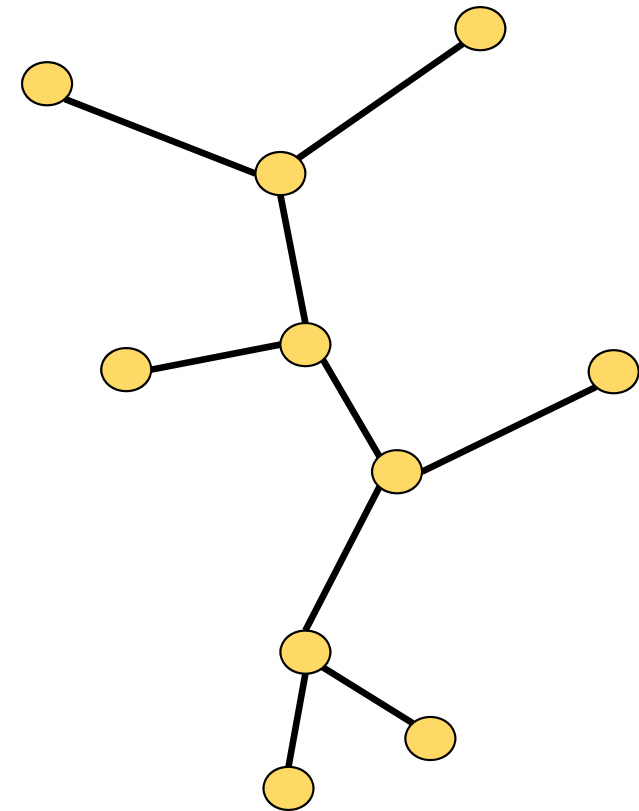
ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V = 全ての点集合

E = 全ての異なる点の対

枝 (i,j) の費用 $= d(i,j)$

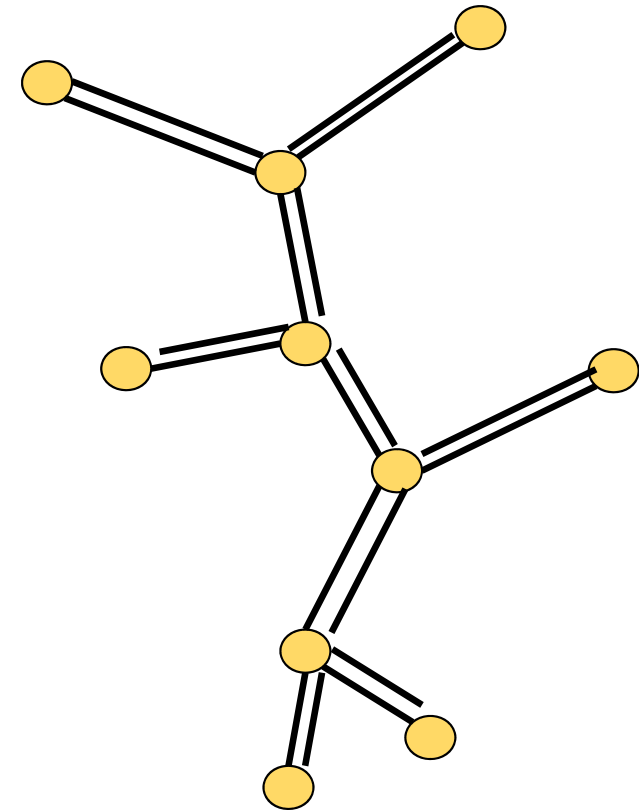
最小全域木の総距離 $\leq$ 最短巡回路長
($\because$ 巡回路から枝をひとつ取り除くと
全域木)



アルゴリズムの近似比解析

ステップ2: 最小全域木の枝を全て二重にする.

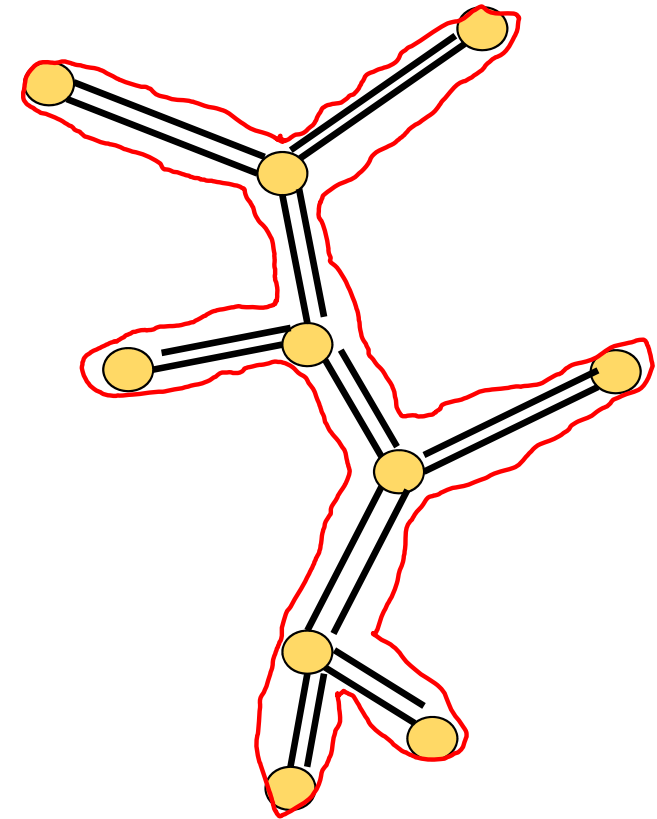
枝の長さの合計
 $\leq 2 \times (\text{最小全域木の長さ})$



アルゴリズムの近似比解析

ステップ3: オイラー閉路を作る.

オイラー閉路の長さ=枝の長さの合計

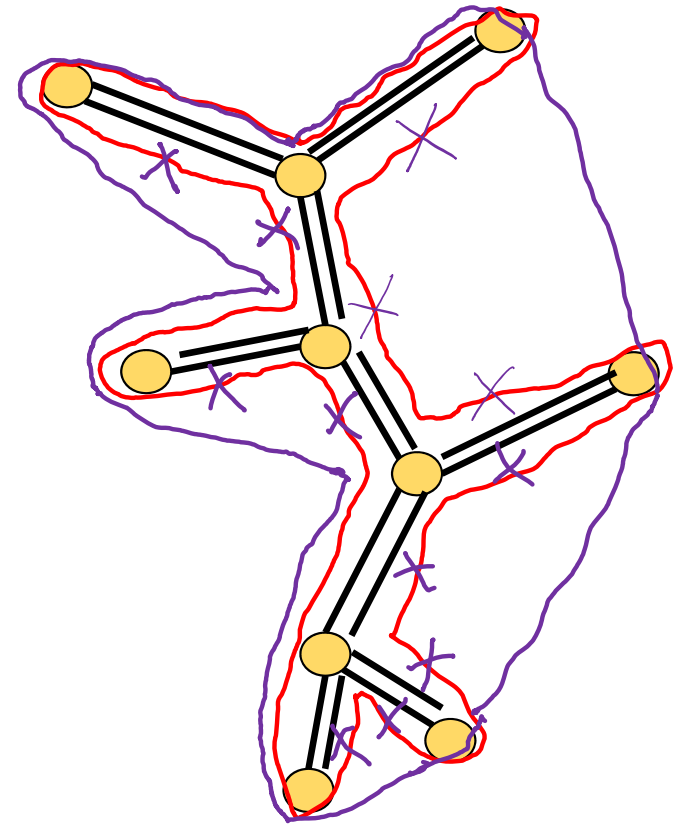


アルゴリズムの近似比解析

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.

三角不等式により,
ショートカットの長さ
 $\leq$ ショートカットされた部分の長さ

よって,
得られた巡回路の長さ
 $\leq$ オイラー閉路の長さ



アルゴリズムの近似比解析:まとめ

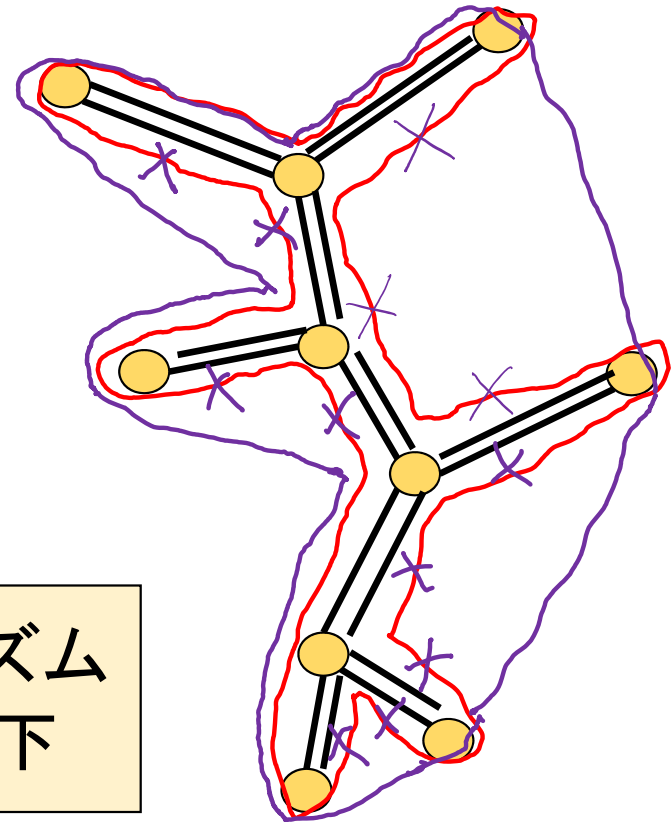
得られた巡回路の長さ

$\leq$ オイラー閉路の長さ

$= 2 \times (\text{最小全域木の長さ})$

$\leq 2 \times (\text{最短巡回路の長さ})$

定理 最小全域木を用いた近似アルゴリズム
の近似比は2以下



最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム

最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム: 準備

最小重み k マッチング問題:

枝数 k のマッチングで枝重みの和が最小のものを求める

最小重み完全マッチング問題:

全ての頂点を繋ぐマッチングで枝重みの和が最小のものを求める

最小重み k (完全)マッチング問題の解き方:

枝重みに -1 を掛けて, 最大重み k (完全)マッチング問題を解けばよい

最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム

ステップ1: 最小全域木を求める

ただし, 次のグラフを考える.

V =全ての点集合

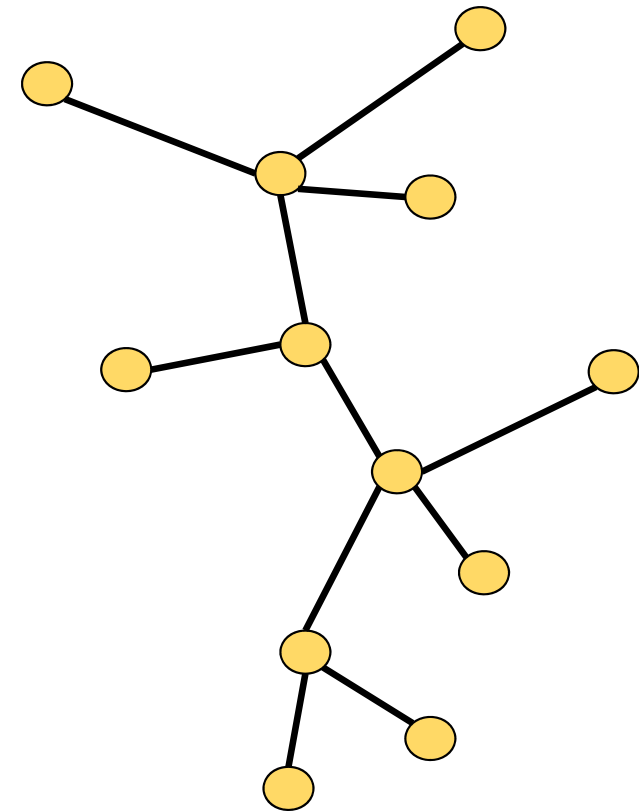
E =全ての異なる点の対

枝 (i,j) の費用 $=d(i,j)$

ステップ2: 奇数次数の頂点のみに
関する最小重み完全マッチングを求める

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム

ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V =全ての点集合

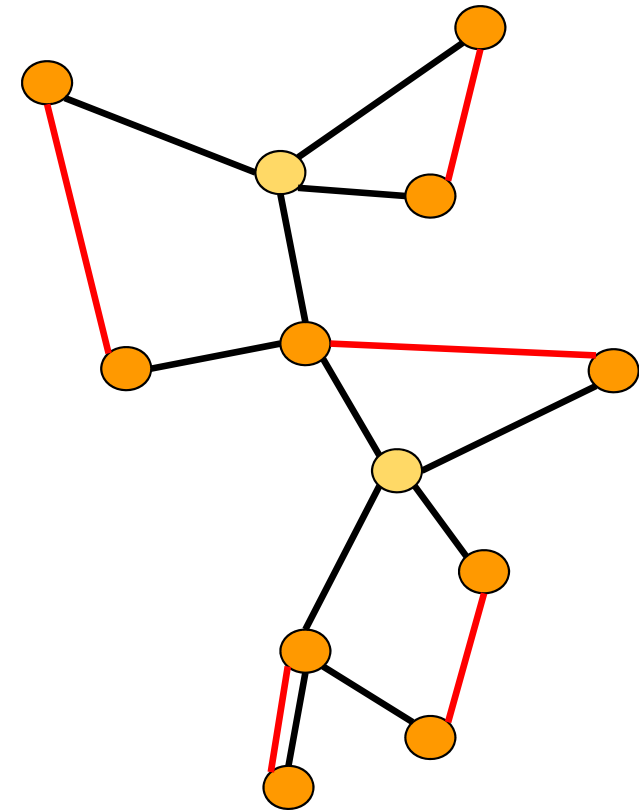
E =全ての異なる点の対

枝 (i,j) の費用 $=d(i,j)$

ステップ2: 奇数次数の頂点のみに
関する最小重み完全マッチングを求める

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



奇数次数の頂点の総数は偶数
→ 完全マッチングが存在する

最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム

ステップ1: 最小全域木を求める

ただし、次のグラフを考える.

V = 全ての点集合

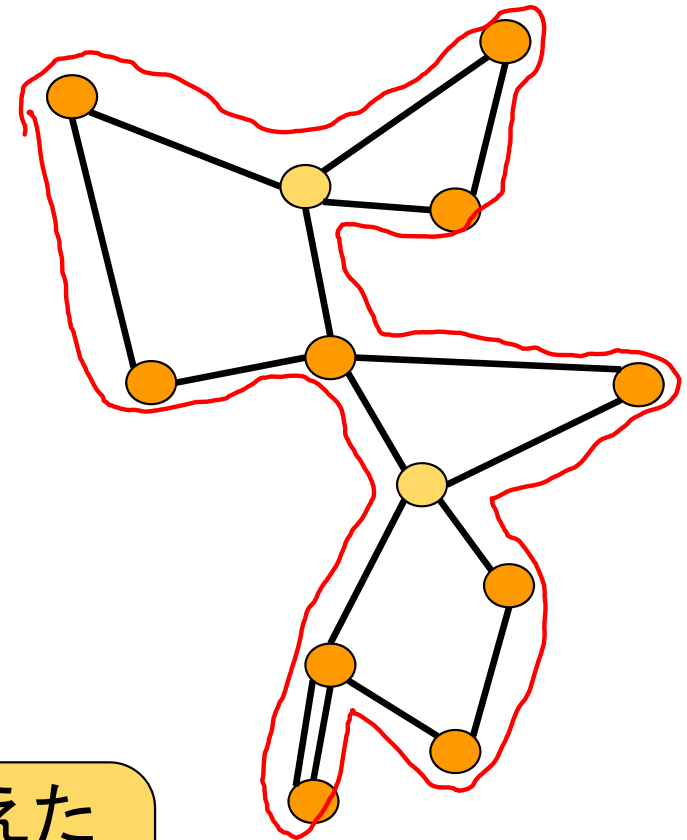
E=全ての異なる点の対

枝(i,j)の費用= $d(i,j)$

ステップ2: 奇数次数の頂点のみに 関する最小重み完全マッチングを求める

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところをショートカットして、巡回路にする.



奇数次数の頂点の次数はいずれも1増えた

→全ての頂点の次数が偶数

→オイラー閉路が存在

最小全域木と最小重みマッチングを用いた1.5近似アルゴリズム

ステップ1: 最小全域木を求める

ただし, 次のグラフを考える.

V =全ての点集合

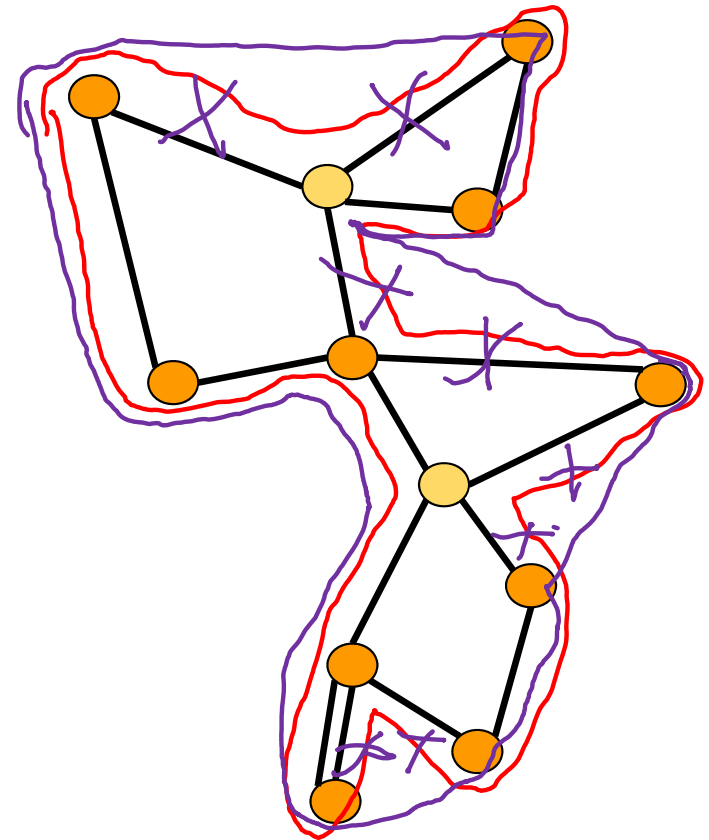
E =全ての異なる点の対

枝 (i,j) の費用 $=d(i,j)$

ステップ2: 奇数次数の頂点のみに
関する最小重み完全マッチングを求める

ステップ3: オイラー閉路を作る.

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.



アルゴリズムの近似比解析

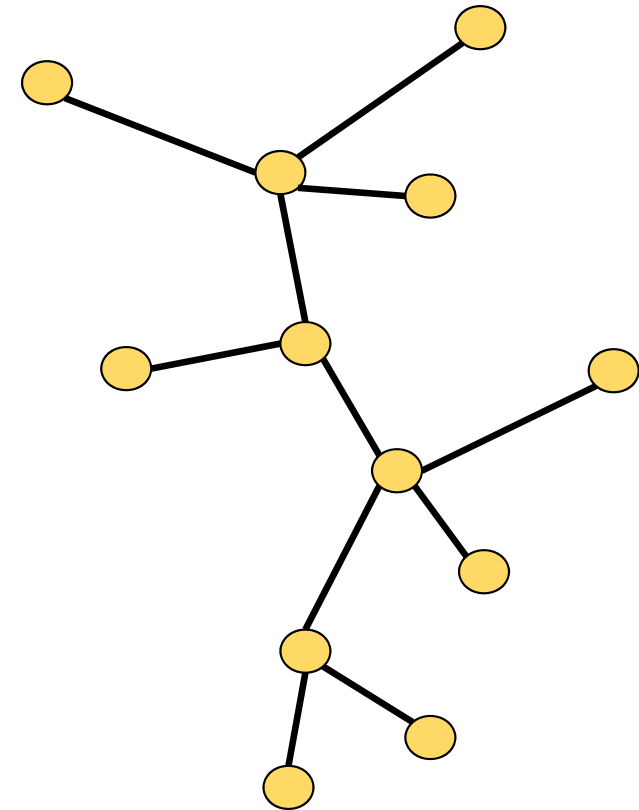
ステップ1: 最小全域木を求める
ただし, 次のグラフを考える.

V = 全ての点集合

E = 全ての異なる点の対

枝 (i,j) の費用 $= d(i,j)$

最小全域木の総距離 $\leq$ 最短巡回路長
($\because$ 巡回路から枝をひとつ取り除くと
全域木)



アルゴリズムの近似比解析

ステップ2: 奇数次数の頂点のみに
関する最小重み完全マッチングを求める

最小重み完全マッチングの総距離
 $\leq 0.5 \times$ 最短巡回路長

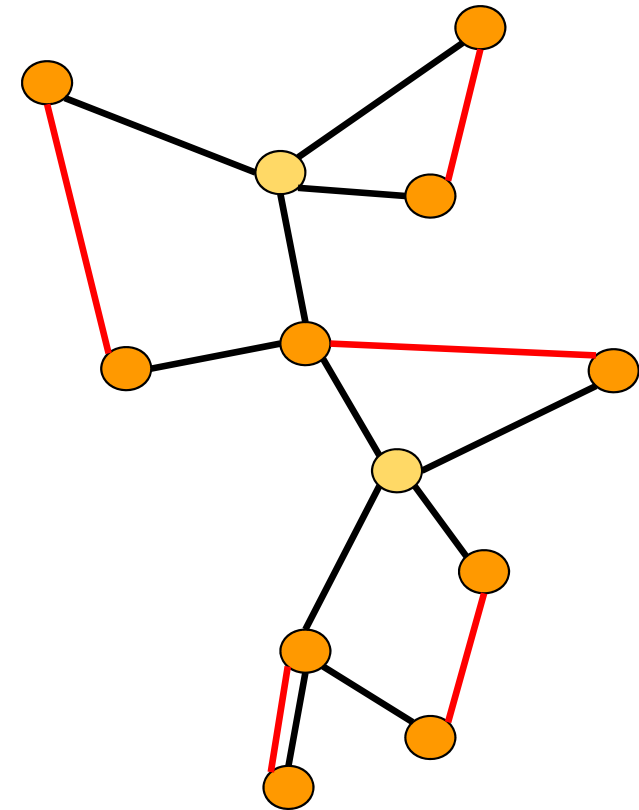
(証明)

最短巡回路長

$\geq$ 奇数次数の頂点を残してショートカット
して得られた閉路の長さ

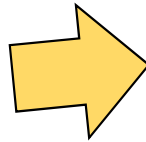
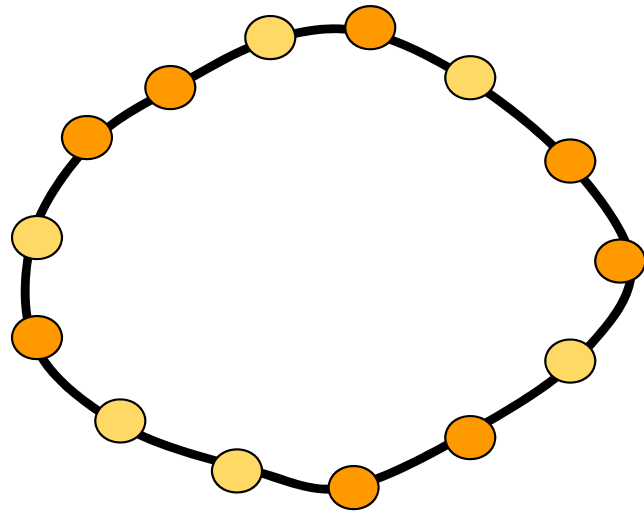
$=$ 奇数次数の頂点に対する2つの
完全マッチングの総距離の和

$\geq 2 \times$ 最小重みマッチングの総距離

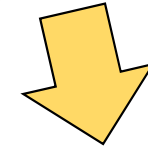
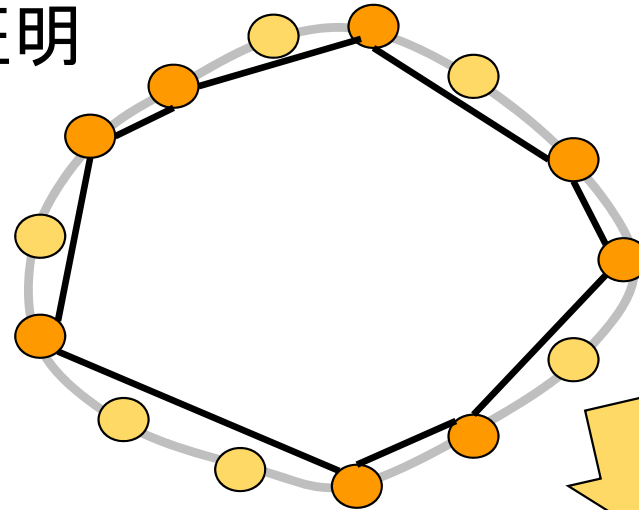


アルゴリズムの近似比解析

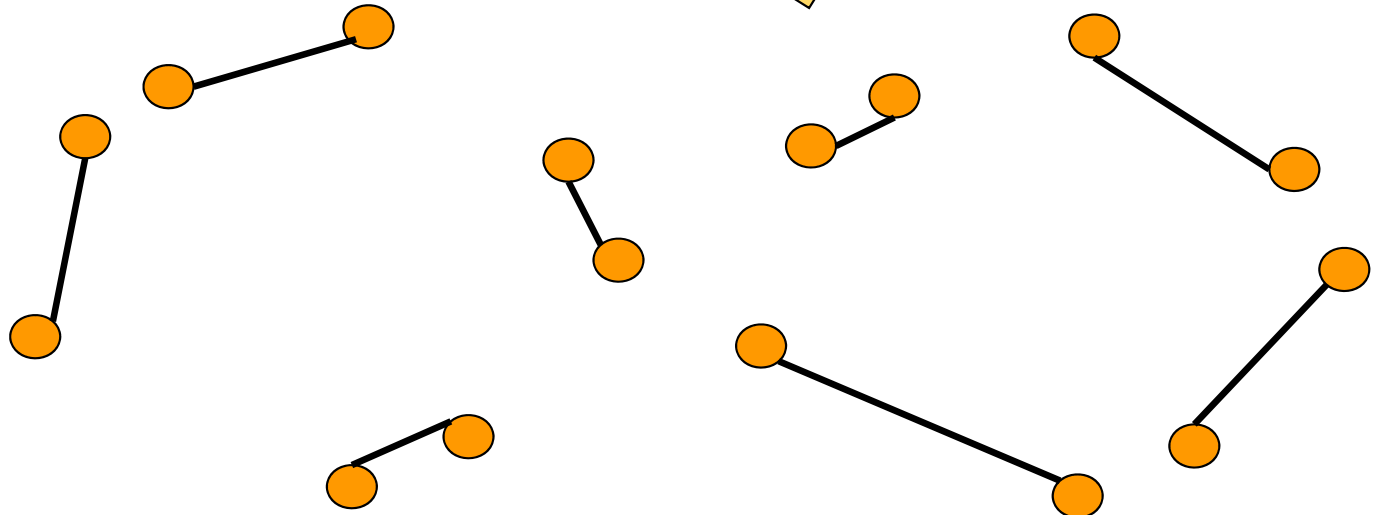
「最小重み完全マッチングの総距離
 $\leq 0.5 \times$ 最短巡回路長」の証明



最短巡回路を,
 奇数次数の頂点を
 残してショートカット



奇数次数の頂点に
 対する2つの完全
 マッチングに
 分解可能

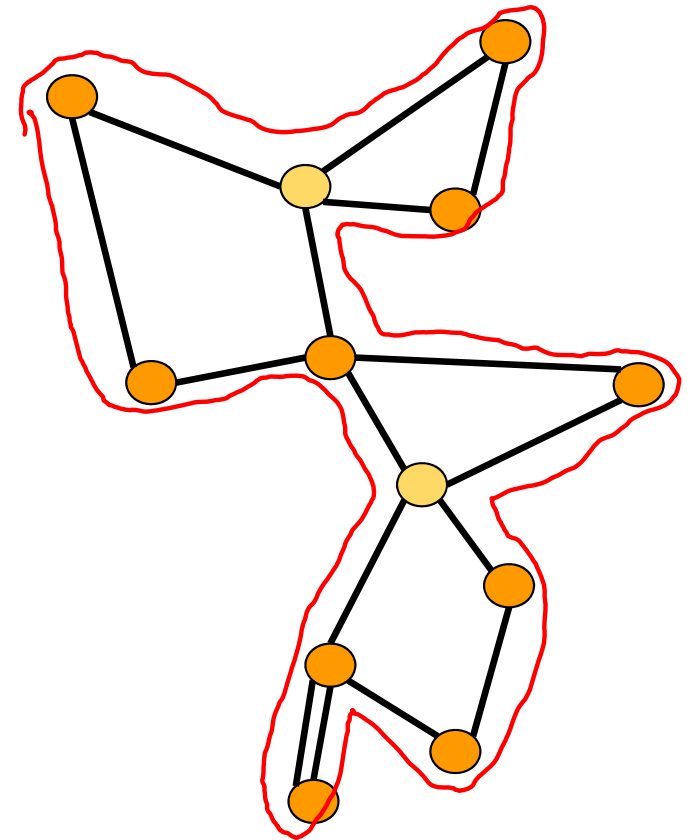


各完全マッチングの総距離 $\geq$ 最小重み完全マッチングの総距離

アルゴリズムの近似比解析

ステップ3: オイラー閉路を作る.

オイラー閉路の長さ
= 最小全域木の総距離
+ 最小重みマッチングの総距離

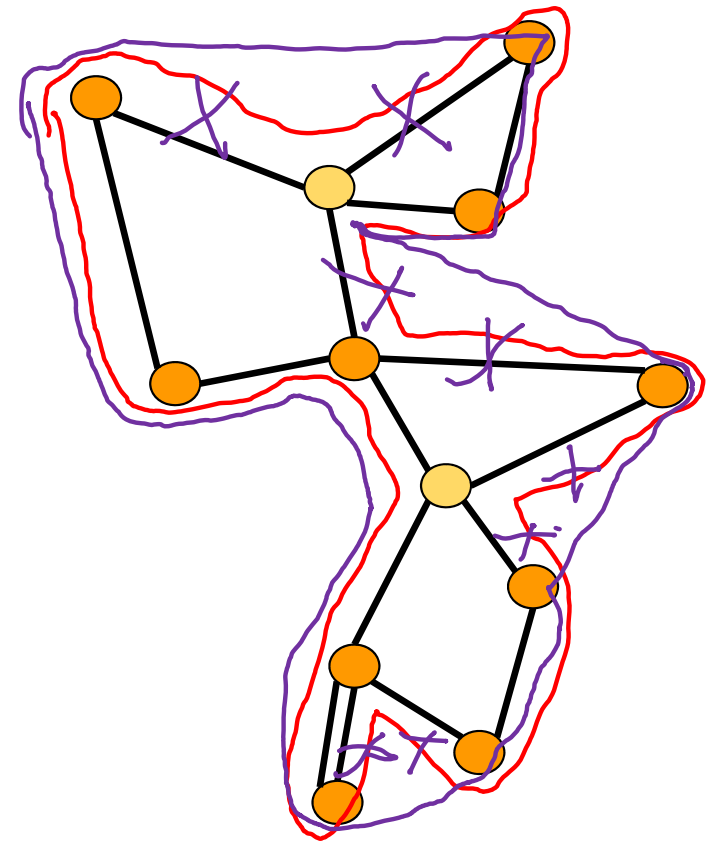


アルゴリズムの近似比解析

ステップ4: オイラー閉路の無駄なところ
をショートカットして, 巡回路にする.

三角不等式により,
ショートカットの長さ
 $\leq$ ショートカットされた部分の長さ

よって,
得られた巡回路の長さ
 $\leq$ オイラー閉路の長さ



アルゴリズムの近似比解析:まとめ

得られた巡回路の長さ
 $\leq$ オイラー閉路の長さ
 $=$ 最小全域木の総距離
 $\quad +$ 最小重みマッチングの総距離
 $\leq$ (最短巡回路の長さ)
 $\quad + 0.5 \times$ (最短巡回路の長さ)
 $= 1.5 \times$ (最短巡回路の長さ)

定理 最小全域木と最小重みマッチングを用いた近似アルゴリズムの近似比は1.5以下

