

$G = (V, E)$ 連結無向グラフ

$C_e \geq 0$ ($e \in E$) 枝コスト

Kruskal (クラスカル) のアルゴリズム

まず枝コストが小さいもの順に枝をなすへ加える

$E = \{e_1, e_2, \dots, e_m\}$

$\hookrightarrow$ ソート $O(m \log m)$

$C_{e_1} \leq C_{e_2} \leq \dots \leq C_{e_m}$

選ばれる枝

$F = \emptyset, H = E$

While ($H \neq \emptyset$) do {

Choose $e \in H$ with minimum C_e

$H \leftarrow H \setminus \{e\}$ (H から e と e を除く)

If $F \cup \{e\}$ has no cycle

$F \leftarrow F \cup \{e\}$

$F \cup \{e\}$ はサイクルがなければ F に e を追加する
追加しない

Output $T^0 = (V, F)$

Lem Output $T^0 = (V, F)$ は全域木

(:) サイクルを含まない $\rightarrow$ OK

連結で $n-1$ の V の頂点に接続しているか?

そうだったとしたら...

$T^0 = (V, F)$ はいくつかの連結成分がある
1つ以上



互いの頂点間に
極大の頂点集合

連結グラフ G の部分グラフ

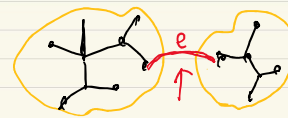
G の中で T^0 と T^0 の異なる連結成分を

必ずパスが存在する

そのパスの中の枝 T^0 の異なる連結成分

をつなぐ枝がある

その枝を T^0 に追加しても サイクルはあられぬ (✖)



もしあられぬなら
そのサイクルの
 e とは T^0 の枝
連結成分をつなぐものがある

その枝 e は アルゴリズム中に F に追加できなかった

$F \cup \{e\}$ は サイクルなり

しかし F は T^0 の枝の部分集合

$\rightarrow T^0$ に追加しても サイクルなり

(✖) と ムジーン

次にしるべきは: T^0 が 最小木

枝コストの総和
が最小

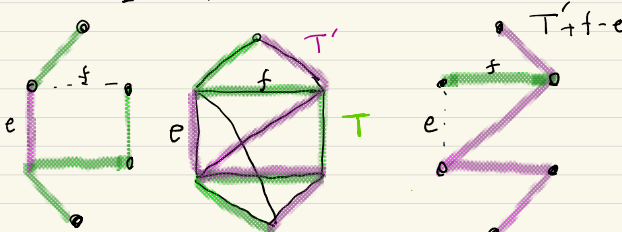
Lem T, T' : 全域木 (グラフ理論的性質)
マトロイド性

任意の $e \in T' \setminus T$ (T' に $\lambda, \lambda, \lambda, \lambda, \lambda$ あり T に $\lambda, \lambda, \lambda, \lambda, \lambda$ あり枝)

に対して ある $f \in T \setminus T'$ (T に $\lambda, \lambda, \lambda, \lambda, \lambda$ あり T' に $\lambda, \lambda, \lambda, \lambda, \lambda$ あり)

$T + e - f, T' + f - e$

も 全域木



(:) $e \in T' \setminus T$

e を T に追加してみる

e の端点 x, y をつなぐ T のパスが存在

(2つあると サイクルができてしまう) $T \cup \{e\} \rightarrow$ ムジーン



e と T のパス
= サイクル

$C(T|e)$: 基本サーキット

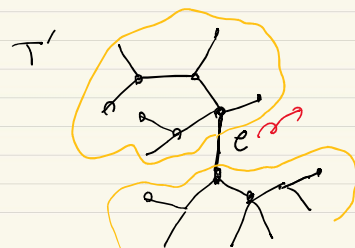
OBS $T + e - f$ ($f \in C(T|e)$)

も 全域木

$T + e$ は サイクル
の一部分
の サイクル

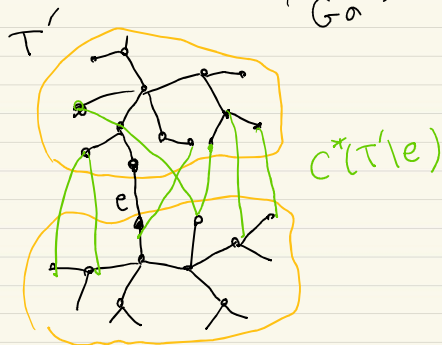
連結成分

T' から e と f をとる $\rightarrow$ 2つの木に分かれる



$C^*(T|e)$: 基本カットセット

e との重なりを消すに 5 分ける
2つの連結成分をまたぐ辺からなる
集合 G の



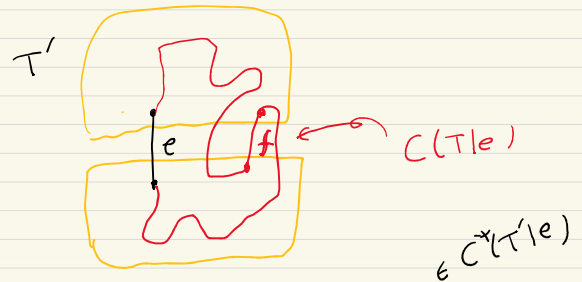
OBS $T' - e + f$ ($f \in C^*(T|e)$)
は全域木

$e \in T' \setminus T$ と T に追加する

$\rightarrow C(T|e)$ 基本カットセット

$e \in T' \setminus T$ の重なり

$\dots e$ とぶつかるサイクル



$\rightarrow C(T|e)$ には e と重なる枝 + T

$T' - e$ の連結成分をまたぐものがある

$T + e - f$ = 全域木

$T' - e + f$ = 全域木

(Kruskal の正当性の証明)

T^0 : output T^* : 最小木 1 本, 2

$|T^0 \cap T^*|$ 最大

12 になるものとしてくる。

帰理法 $T^0 \neq T^*$ と仮定 (Goal: $T^0 = T^*$)

$e \in T^0 \setminus T^*$

アルゴリズムで最初に選ばれた T^* 以外の枝

$f \in T^* \setminus T^0$:

$T^0 - e + f$ = 全域木

$T^* + e - f$ = 全域木

全域木の枝数 = $|V| - 1$

別のやり方 e を選ぶときは f を選んでもサイクルは 5 分ける

$C_e \leq C_f \leftarrow f$ なら e が選ばれた
の?"

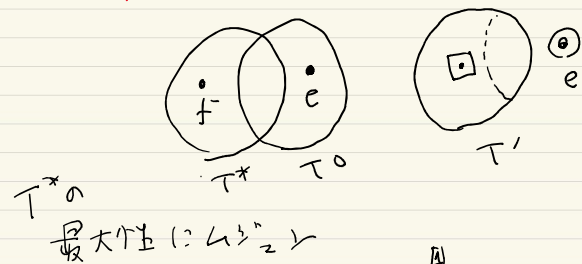
$T' = T^* + e - f$ のコストは?

$C(T') = C(T^*) + C_e - C_f \leq C(T^*)$

$\sum_{e' \in T'} C_{e'} \leq \sum_{e' \in T^*} C_{e'} \rightarrow C(T') = C(T^*)$
 $\rightarrow T' = \text{最小木}$

しかし, $|T' \cap T^0| > |T^* \cap T^0|$ と分る?

"
 $|T^* \cap T^0| + 1$



T^* の
最大値に 1 だけ増える

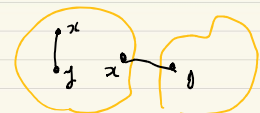
• 計算量の評価

$n = |V|$
 $m = |E|$
 $m \leq n^2$

• ソートプログラムの時間 $\dots O(m \log m) = O(m \log n)$

• $F \cup e$ はサイクルを作るかどうか

$\leftrightarrow e$ の異なる連結成分をまたぐかどうか
 xy



Union-Find テータ 挿入 によって while 全体

$$O(n \log n)$$

$$\leadsto O(m \log n)$$

・ソートアルゴリズム リストの長さ n

2 3 5 1 6 4 13

小さいもの順に
並べかえる

1 1 2 3 3 4 5 6

$$O(n^2)$$

バブルソート

右から左へ 比較 合う 数と交換して

a, b

もし、 $b < a$ なら

$\begin{matrix} a, b \\ \swarrow \searrow \\ b, a \end{matrix}$

右側に行ったら 左からやり直す

2 3 4 1 6 4 13



2 3 4 1 6 1 4 3



2 3 4 1 1 6 4 3



2 3 1 4 1 6 4 3



2 1 3 4 1 6 4 3



1 2 3 4 1 6 4 3

3 6 4

1 1 2 3 4 3 6 4

4 6