

ダイクストラ (Dijkstra) のアルゴリズム

Sからsの最短経路が確定した点全体

$S := \{s\}, d(s) := 0, P_s = \phi$

while $S \neq V$ do {

$d(v)$: sからvへの最短経路

P_v : sからvへの最短経路

$h(v) := \infty$

• For each $v \in V \setminus S$, set

$h(v) := \min \{ d(u) + le \mid e = (u,v), u \in S \}$

• Choose $v \in V \setminus S$ such that

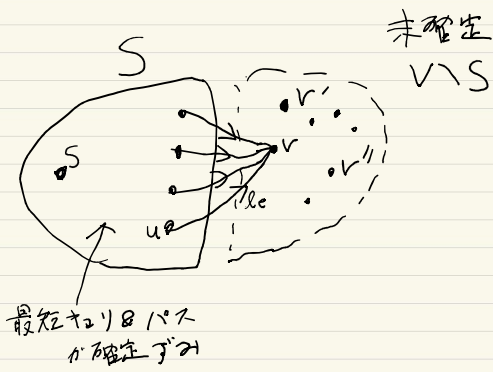
$h(v)$ is minimum

• $S \leftarrow S \cup \{v\}, d(v) := h(v)$

$P_v := P_u \cup e$

$\hookrightarrow h(v)$ を達成する枝

}



Prop アルゴリズムは正しい

つまり $u \in S, P_u = (s,u)$ - 最短経路
 $d(u)$ = 最短経路

IS11材料

⑤ 帰納法

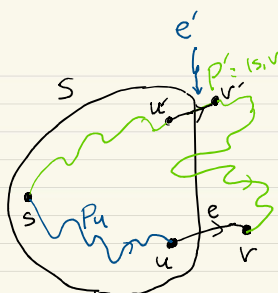
$S = \{s\}$ のとき

アルゴで v が S に追加されたとき

示したいこと: 今の (s,v) - パス P'_v は正しい

$l(P_v) = l(P_u) + le \leq l(P')$

$\parallel e = (u,v)$
 $d(u)$



P'_v は v に到達する前に一度 S を出る必要がある

P'_v が S を最初に出る瞬間に着目する

$l(P') \geq d(u') + le'$

$e' = (u',v')$

アルゴリズムより

$e \rightarrow \geq d(u) + le$

計算量の分析

h の更新の方向: 新しく確定した頂点 v

から出ている枝 $e = (v,w)$

でつながっている頂点 w の $h(w)$

を更新する必要がある

--- $d(v) + le$ の計算と

$h(w)$ の比較

と V から出ている各枝に対して行う

頂点が追加されるまで: $O(|\text{その頂点から出る枝}|)$

全体で $\sum_{v \in V} O(|v \text{ から出る枝}|)$

$\parallel$

$O(|E|)$

$h(v)$ が最小になる

v を選ぶ方向: $O(|V|)$

$\hookrightarrow O(1)$

更新の方向 $O(\log n)$

Whileループがまわる回数: $|V|$

--- $O(|V|^2 + |E|) \sim O(|V||E|)$

工夫すると $O(n \log m)$ $n = |V|$
 $m = |E|$
 テーブルというデータ構造

$e \rightarrow$ 25にほく...
 所算がついている。

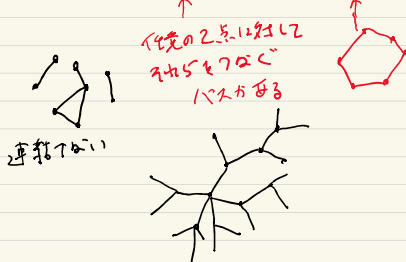
最小木問題

無向グラフ $G = (V, E)$

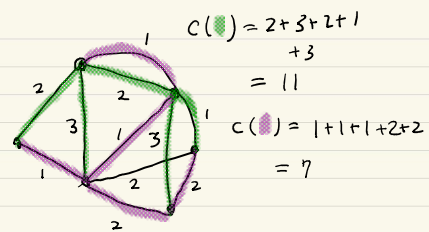
コスト $C_e \geq 0 (e \in E)$

全域木 T のコスト $c(T) = \sum_{e \in T} C_e$

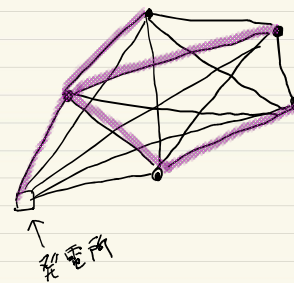
$\Rightarrow$ 木 $\Leftrightarrow$ 連結グラフ サイクルをぶくまぬい



G の全域木 $\Leftrightarrow$ 部分グラフであり、木、すべての頂点と接続



1920年代ごろ



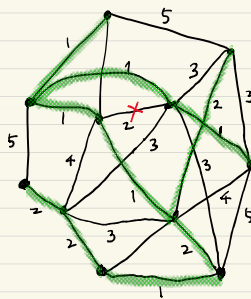
図

都市

発電所から
 電線をはいて
 すべての都市に
 電力を供給したい

つかう電線の長さを
 みだかくなかい

すべての都市と発電所を頂点とした完全グラフ
 木のコストとして
 最小値をとる。 $\Rightarrow$ すべての頂点内に接続



緑 = 全域木 $\rightarrow$ 最小木