

グラフとネットワーク (第9回)

<http://coconut.sys.eng.shizuoka.ac.jp/gn/06/>

安藤和敏 (静岡大学工学部)

2007.01.16

2.1. 木と道

木は p. 13 において、「閉路を含まない連結なグラフ」として定義された。

グラフ $G = (V, A)$ が与えられたときに、 G の全域部分グラフで木であるようなものを G の木と呼ぼう。つまり G の木とは、ある $T \subseteq A$ に対して $H = (V, T)$ と表されるグラフである。

さらに、 G の枝の部分集合 $T \subseteq A$ に対して、部分グラフ $H = (V, T)$ が G の木になるときに、 T を G の木と呼ぶ。

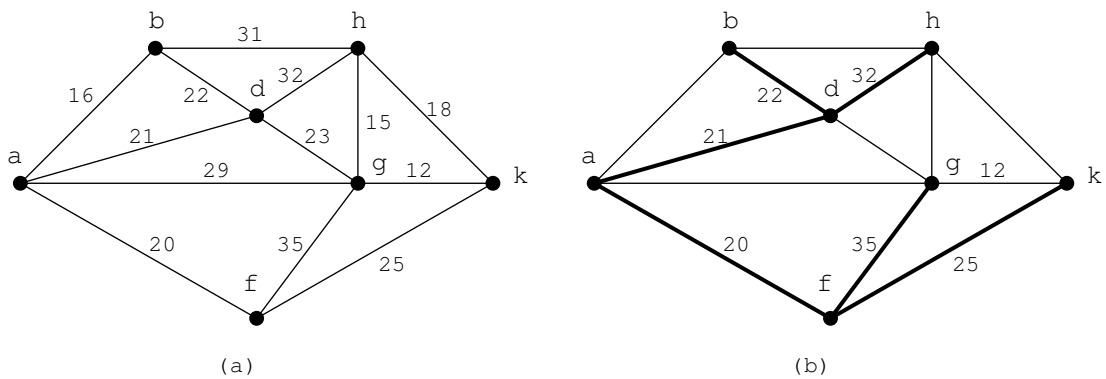


図 2.1: (a) グラフ G と $w: A \rightarrow \mathbb{R}$; (b) G の木 T (太字の枝)

2.1.1. 最小木問題

連結な無向グラフ $G = (V, A)$ と枝集合上の重み関数 $w: A \rightarrow \mathbb{R}$ が与えられているとする。 G の木 $T \subseteq A$ に対して

$$w(T) = \sum_{a \in T} w(a) \quad (2.1)$$

で定義される $w(T)$ を T の重み (weight) という。 $G = (V, A)$ が与えられたときに、重みが最小である木を見出す問題を最小木問題 (minimum-tree problem) と呼ばれる。

最小木問題は貪欲アルゴリズム (greedy algorithm) と呼ばれるつぎの手順で解くことができる。

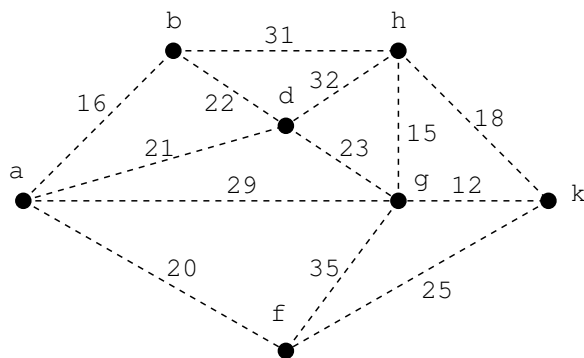
Algorithm 1 貪欲アルゴリズム (クラスカル (Kruskal) のアルゴリズム)

Require: 無向グラフ $G = (V, A)$, 重み関数 $w: A \rightarrow \mathbb{R}$.

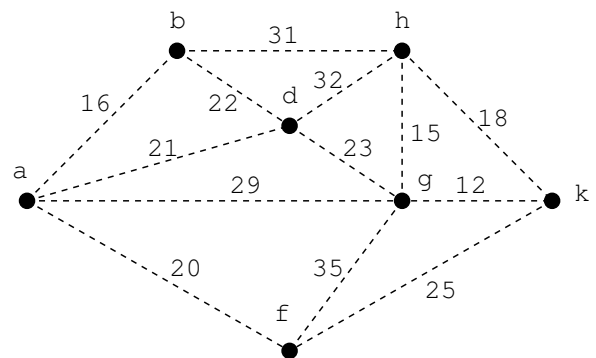
Ensure: G の最小木 T .

- 1: $T \leftarrow \emptyset$.
- 2: 現在の T が木ならば停止 (T は最小木). そうでなければ, $T \cup \{a\}$ がサーキットを含まないような枝 $a \in A \setminus T$ のうちでその重み $w(a)$ が最小であるものを1つ選び, それを \hat{a} とおく.
- 3: $T \leftarrow T \cup \{\hat{a}\}$ として, 2へ戻る.

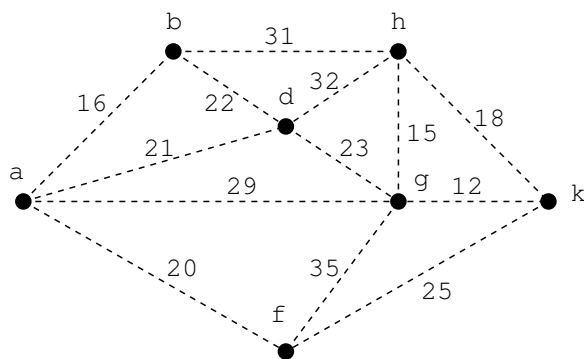
例として, 図 2.1(a) の $G = (V, A)$ と w を入力として, 貪欲アルゴリズムを動かしてみよう. (学生諸君は, 私の板書を見ながら, 図に T を書き込みなさい.)



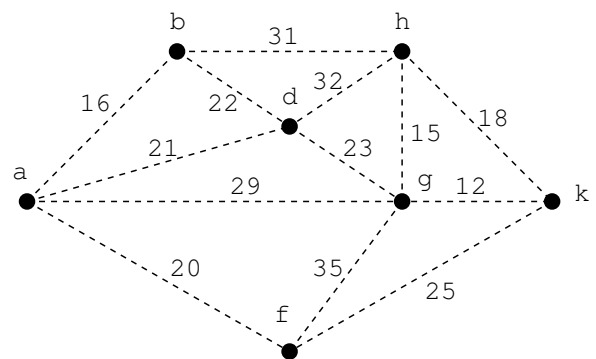
1回目の Step 3 の終了時の T



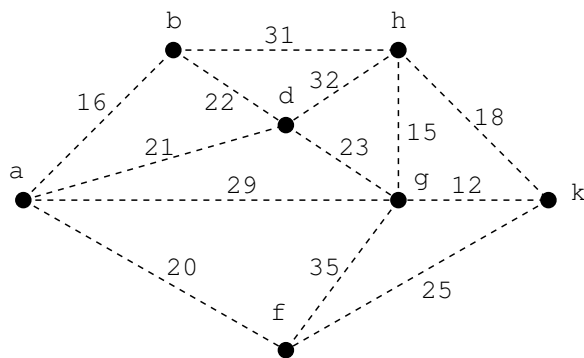
2回目の Step 3 の終了時の T



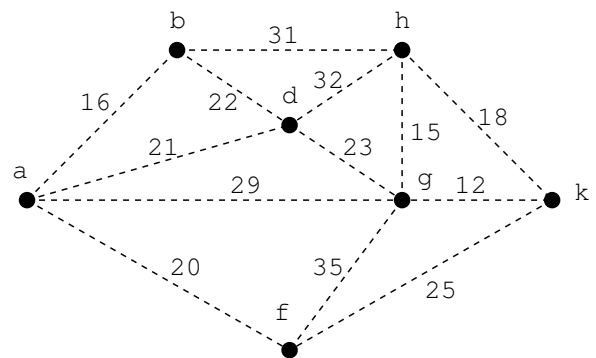
3回目の Step 3 の終了時の T



4回目の Step 3 の終了時の T



5回目の Step 3 の終了時の T



6回目の Step 3 の終了時の T

グラフ $G = (V, A)$ の木 T と枝 $a \in A \setminus T$ に対して, $T \cup \{a\}$ はちょうど一つの閉路を含む. この閉路を $C(T|a)$ と表し, T と a に関連する基本サーキットと呼ぶ.

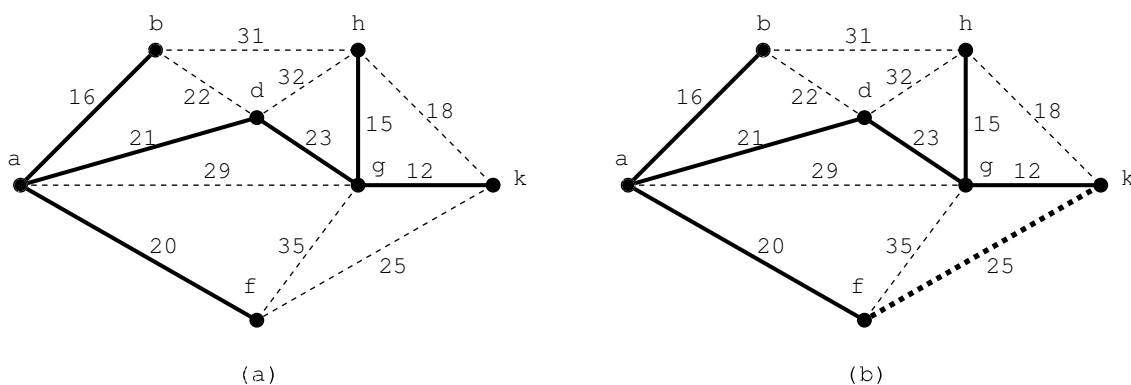


図 2.2: (a) グラフ G の木 T (実線の枝からなる) と $w: A \rightarrow \mathbb{R}$; (b) T と $a = (f, k)$ に関連する基本サーキット $C(T|a) = \{(a, f), (f, k), (k, g), (g, d), (d, a)\}$

定理 2.1: グラフ $G = (V, A)$ の木 $T \subseteq A$ が重み $w: A \rightarrow \mathbb{R}$ に関する最小木であるための必要十分条件は, 各 $a \in A \setminus T$ とそれに関連する基本サーキット $C(T|a)$ 上の任意の枝 a' に対して

$$w(a') \leq w(a) \tag{2.5}$$

が成り立つことである.

最小木問題はヤルニーク-プリム (Jarník-Prim) のアルゴリズムと呼ばれるつぎの手順によっても解くことができる.

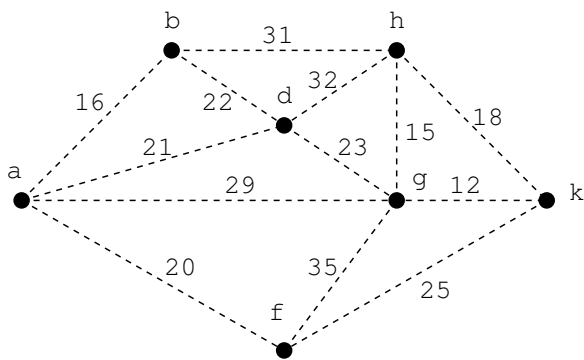
例として, 再び図 2.1(a) の $G = (V, A)$ と w を入力として, ヤルニーク-プリムのアルゴリズムを動かしてみよう. (学生諸君は, 私の板書を見ながら, 図に T を書き込みなさい.)

Algorithm 2 ヤルニーク-プリム (Jarník-Prim) のアルゴリズム

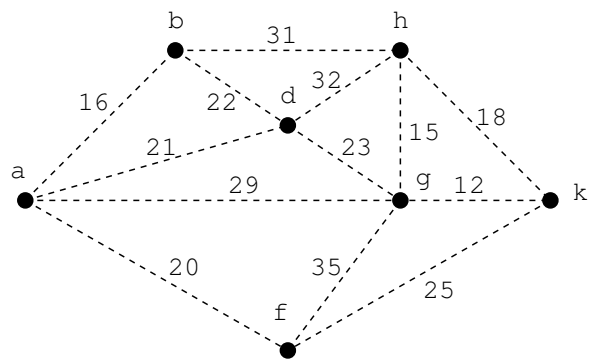
Require: 無向グラフ $G = (V, A)$, 重み関数 $w: A \rightarrow \mathbb{R}$.

Ensure: G の最小木 T .

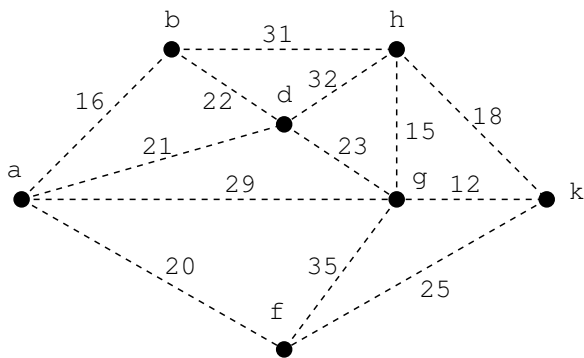
- 1: 任意な点 $v \in V$ を選び, $U \leftarrow \{v\}$, $T \leftarrow \emptyset$ とおく.
- 2: 現在の T が G の木 (すなわち $U = V$) ならば停止 (T が最小木である). そうでなければ, U と $V \setminus U$ を結ぶ枝 a のうちでその重み $w(a)$ が最小であるものを1つ選ぶ. 選ばれた枝を \hat{a} とおいて, \hat{a} の $V \setminus U$ 側の終点を u とおく,
- 3: $U \leftarrow U \cup \{u\}$, $T \leftarrow T \cup \{\hat{a}\}$ として, 2へ戻る.



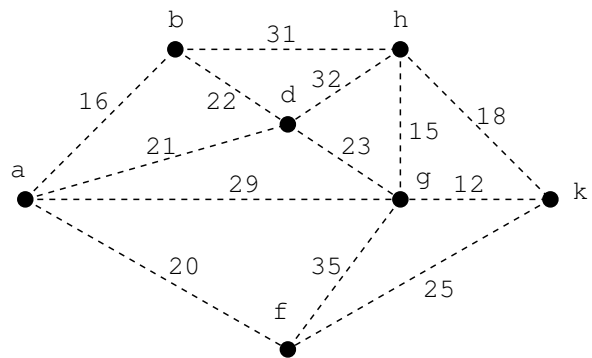
1回目の Step 3 終了時の T



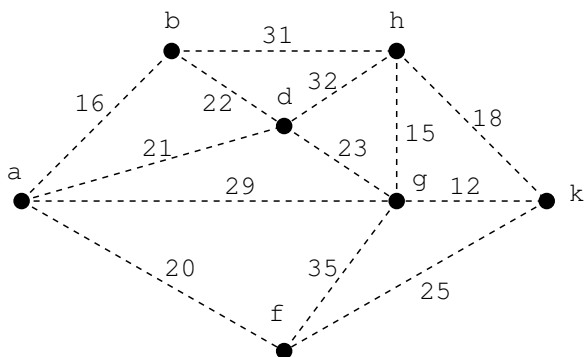
2回目の Step 3 終了時の T



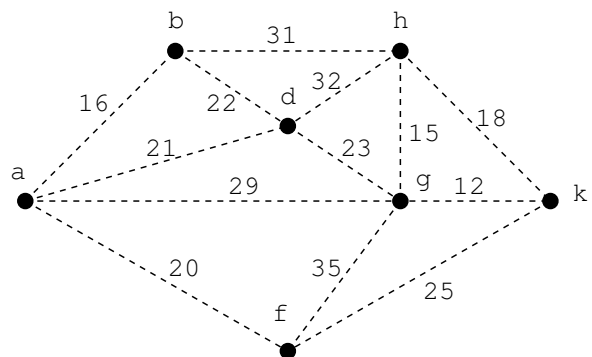
3回目の Step 3 終了時の T



4回目の Step 3 終了時の T



5回目の Step 3 終了時の T



6回目の Step 3 終了時の T