# 最大全域木アルゴリズム

原田 大瑚

2024/03/31

## 1　最大全域木

$G$ が連結グラフの時、$G$ の全域木 $T$ の重み、すなわち $T$ に含まれる辺の重みの総和を $\omega(T)$ と表す。$\omega(T)$ が最大の全域木を最大全域木、最小の全域木を最小全域木という。

## 2　クラスカルのアルゴリズム

### 2.1　アルゴリズム

ネットワークの最大全域木を求める最適化問題「最大全域木問題」は以下のように記述できる。入力：連結グラフ $G$、重み関数 $\omega : E(G) \to R$ 質問: ネットワーク $(G, \omega)$ の最大全域木を 1 つ示せ。

クラスカルのアルゴリズムは最大全域木問題を解く多項式時間のアルゴリズムである。入力：連結グラフ $G$、重み関数 $\omega : E(G) \to R$（ただし、$|V(G)| = n$、$|E(G)| = m$ とする）出力：ネットワーク $(G, \omega)$ の最大全域木 T

ステップ 0: 辺 $e_1$ の重みを $\omega(e_1)$ と記述する。$E(G)$ の辺の重みを降順に並べる。$(\omega(e_1) \geq \omega(e_2) \geq ... \geq \omega(e_m))$ ステップ 1:$i = 1, X = \emptyset$ とするステップ 2:$G\langle X \cap \{e_i\}\rangle$ が閉路に含まない場合、$X = X \cup \{e_i\}$ ステップ 3:$|X| = n-1$ ならば、$G\langle X\rangle$ を最大全域木 $T$ として出力する。ステップ 4:$i = i + 1$ として、ステップ 2 に戻る。

つまり重さが重い順に辺を追加していく、ただし閉路が発生してしまうときは例外的に追加しないようにするということである。

このアルゴリズムを応用して、辺の重みを昇順 $(\omega(e_1) \leq \omega(e_2) \leq ... \leq \omega(e_m))$ に並べることで”最小全域木問題” 入力：連結グラフ $G$、重み関数 $\omega : E(G) \to R$ 質問: ネットワーク $(G, \omega)$ の最小全域木を 1 つ示せ。も解くことができる。

### 2.2　時間計算量

ステップ 0 は併合整列アルゴリズムを用いれば $O(m \log m)$ 時間で実行できる。ステップ 1,3,4 は $O(1), O(n), O(1)$ 時間で実行できる。ステップ 2 は $G\langle X \cup \{e_i\}\rangle$ が閉路を含むかどうかを判定する。$e_i = (u, v)$ としたとき $G\langle X\rangle$ に $(u, v)$ 路がすでに存在した時、$G\langle X \cup \{e_i\}\rangle$ が閉路を含むので、$G\langle X\rangle$ に $(u, v)$ 路が存在するか否かで $G\langle X \cup \{e_i\}\rangle$ が閉路を含むかどうかを判定することができる。これは深さ優先探索や幅優先探索を用いて $O(m + n)$ 時間で判定できる。しかし、合併発見手法を用いれば $O(m \log m)$ で実行でき、全体の時間計算量は $O(m \log m)$ にすることができる。

## 3　合併発見手法

### 3.1　手法

合併発見はグラフ $G\langle X\rangle$ に $(u, v)$ 路が存在するか否かを効率よく判別する手法である。合併発見は 2 つの集合を合併する”手続き合併”と集合の代表を求める”手続き発見”からなる。全体の要素を $n$ としたとき、この手法は $O(\log n)$ 時間で実行できる。合併発見では集合は集合に含まれる要素を点とする根付き木で表現され、$(u, v)$ 路が存在するとき、$u$ に対応する点と $v$ に対応する点は同じ根付き木に属する。また、根付き木の根に対応する要素が集合の代表となる。

### 3.2　手続き発見

手続き発見では指定された点から根までたどり、集合の代表を特定する。集合の代表が同じか否かで属する集合が同じか否かを判定できる。

### 3.3　手続き合併

手続き合併は二つの要素が指定された時、手続き発見により、二つの要素が異なる集合に属するとした場合より大きい方の集合に小さい方の集合を従属させる。また、根付き木で表現される場合では小さい方の集合に対応する根付き木の根を大きい方の集合に対応する根付き木の根の子にする。

### 3.4　時間計算量

前述した通りこの手法は $O(\log n)$ で実行できる。これは手続き発見では指定された点から根までたどる為、木の高さが $k$ の時 $O(k)$ 時間で実行できるのが分かる。また、要素数が小さな集合に対応する根付き木の根を他方の根の子とすることで、根付き木の高さを高々$\log_2 n$ できる為実行時間は $O(log_2 k)$ となる。根付き木の高さを高々$log_2 n$ となるのは、$n$ 点から成る根付き木 $T$ として、$|V(T)| >= |V(T$ の部分木$)|$ が成り立つと T の高さは $log_2 n$ に成る為である。そのため、手続き合併では、要素数が小さな集合に対応する根付き木の根を他方の根の子としなければならない。