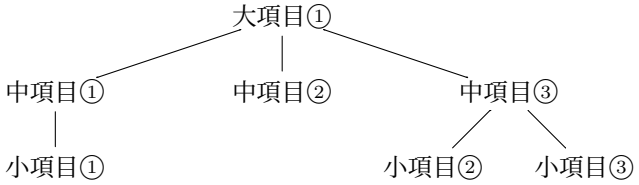
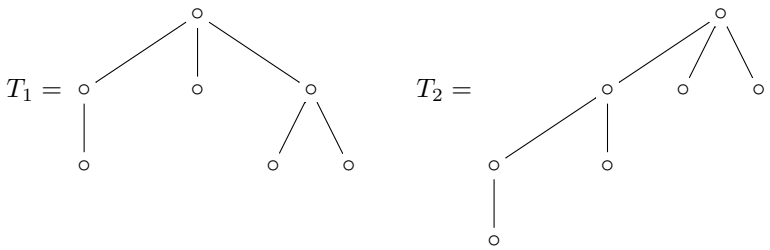
13 離散グラフ理論 （2） –根付き木と幅優先探索アルゴリズム–

• 13-1 : 根付き木

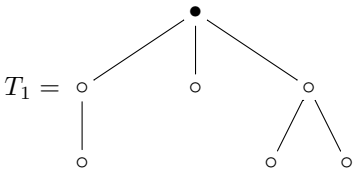
多量のデータを整理するとき, それらのデータをその相互関係に基づいて項目ごとに分類することは現実的によくある話である. その際, 大項目, 中項目, 小項目といったように大きな括りで分けておいて, それらを更に細分してゆく. このような状況はすなわち, 最上位の一点から下に向かって分岐していく木状にデータを分類するということである.



これは, 木の頂点に分類名を置き, 辺で相互関係, 頂点の上下配置によって分類の階層を与えているように捉えることができる. 上の状況では, 木の構造を抜き出すと T_1 のようになっている. しかし, 通常, 無向グラフを図示するときには頂点の配置は気にしないので, T_1 と同じ木を T_2 のように図示することも可能である.



こうしてしまえば, もとの大項目, 中項目, 小項目といった関係は崩れてめちゃくちゃである. そこで, このような状況を回避するために, 「一番上位にあるべき頂点を指定する」ことを考えよう. すなわち, もとの分類問題で「大項目①」のところにある頂点は他のどの頂点よりも上位である, というふうに指定する. この指定された頂点を • で表示して, これが最も上位にある頂点であり, 図示するときは最も上にかく, というふうに定めれば頂点の階層の関係をきちんと反映できているだろう.



この, 木においてただ一つ特別に指定された頂点を **根** と呼び, 根が指定された木を **根付き木** と呼ぶ. 根が v であるような根付き木 $T = (V, E, \psi)$ を組 (T, v) や (V, E, ψ, v) と表す.

さて, 根付き木は有向木として表すこともできる. 根が v であるような根付き木 (V, E, ψ, v) があったとき, **命題 12-4** から v とは異なる任意の頂点 $x \in V$ に対して, v から x への道

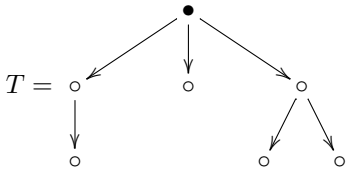
$$(v, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, x), \qquad v_1, \ldots, v_{n-1} \in V,\ e_1, \ldots, e_n \in E$$

がただ一つ存在する. このとき, 辺 e_i の向きを

$$v \xrightarrow{e_1} v_1, \quad v_{i-1} \xrightarrow{e_i} v_i, \quad v_{n-1} \xrightarrow{e_n} x$$

担当：茨城大学 情報科学領域　宮本 賢伍

となるように指定する．すなわち, 各辺の向きを「根から遠ざけるように定義する」ことで, 根付き木は有向木とみなすことができる．

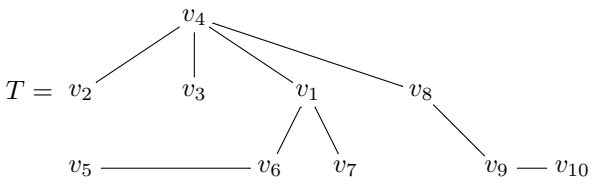


この同一視の方では, 「根付き木」における「根」の概念と「有向木」における「根」の概念は一致する．そこで, 以後, 根付き木を断りなく有向木とみなす．根付き木における **子**, **葉**, **子孫** は T を有向木とみなしたときの子, 葉, 子孫と定義する．また, 頂点 $x \in V$ の **深さ** とは, 根から x へのただ一つの道 w の長さをいう．

> **復習.** 有向木 $T = (V, E, s, t)$ において, 頂点 $v \in V$ に対して, v を始点とする矢印の終点となる頂点を, v の子という．子を持たない頂点を葉という．頂点 v の子孫とは, v もしくは v から道があるような頂点をいう．

定義 13.1. 根付き木 (T, v) について, 頂点の深さの最大値を T の **高さ** と呼ぶ.

レポート 13-1　次の根付き木 (T, v_1) の高さを求めよ．また, 葉を全て列挙せよ．

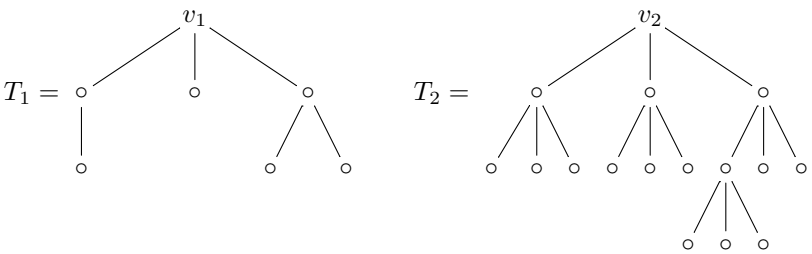


- **13-2 : 正則 m 分木**

定義 13.2. 根付き木 (T, v) は,

- 葉を除く全ての頂点の子の数が高々 m 個であるとき, (T, v) は **m 分木** と呼ばれる．
- 葉を除く全ての頂点の子の数がちょうど m 個であるとき, (T, v) は **正則 m 分木** と呼ばれる．

例 13-1　以下の根付き木 (T_1, v_1) は 3 分木であり, (T_2, v_2) は正則 3 分木である．



正則 3 分木である根付き木 (T_2, v_2) において, 葉の数は 11 であり, 葉ではない頂点の数は 5 である．このとき,

$$(3 - 1) \times 5 = 11 - 1$$

という関係が成り立つ．このような, 正則 m 分木に関して

$$(m - 1) \times (\text{葉ではない頂点の数}) = (\text{葉の数} - 1)$$

という関係式は正則 m 分木の特徴である．つまり, 次の主張が成り立つ．

命題 13.3. 正則 m 分木 (T, v) の葉の数を n とする．このとき, T の葉ではない頂点の数を k とすれば,

$$(m-1)k = n-1$$

が成り立つ.

証明. 正則 m 分木 (T, v) を次のように捉えよう．1 回の試合で m 人の人が勝負をして, そのうちの 1 人だけが勝ち上がるトーナメント戦を想定して, 優勝する 1 人を決める．つまり, 1 試合で $m-1$ 人の人が敗北する．このとき, 葉ではない頂点 k の数は, このトーナメントの全試合数に相当する．すると, k 回の試合で, $(m-1)k$ 人の人が敗北し, これは優勝者以外の人の数である．試合に参加する人の数は葉の数と n 等しいから,

$$(m-1)k = n-1$$

が成り立つ. □

レポート 13-2　正則 5 分木の葉の数が 77 であるとする．このとき, 根でも葉でもない頂点は何個あるか求めよ.

- **13-3 : 幅優先探索 (BFS) アルゴリズム**

グラフとして整理されたデータを, 始点から近い順で走査し, 必要なデータを求める方法の一つが「幅優先探索 (BFS) アルゴリズム」である.

アルゴリズム 13.4 (BFS アルゴリズム)**.** 始点から各頂点への距離を求めるアルゴリズムを記述する.
(**Input**)　連結なグラフ $G = (V, E, \psi)$ と始点 $s \in V$.
(**Output**) G の全域木 T と各頂点 v へのラベル $d(v)$.

`step.0`： 任意の $v \in V$ に対して, $d(v) = *$ とラベルをつける．その後, $d(s) = 0$, $i = 0$, $E(T) = \varnothing$ とする.

`step.1`： $d(u) = i$ である頂点 u に対して, 頂点 u と接続している頂点 u' であって, $d(u') = *$ であるような頂点 $u' \in V$ をすべて探索して `step.2` へ．そのような頂点が存在しないときは終了する.

`step.2`： 頂点 u' が探索されたとき, $d(u') = i+1$ とする．その後, $E(T)$ を

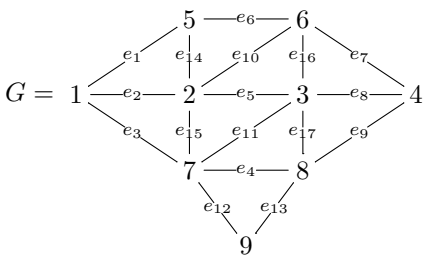
$$E(T) \cup \{e \in E \mid \psi(e) = \{u, u'\},\ d(u) = i,\ u' \text{ は } \texttt{step.1} \text{ で探索した頂点 }\}$$

に更新する．その後, 他に $d(u) = i$ となるような頂点 u があれば `step.2` へ．そうでないならば `step.3` へ.

`step.3`： i を $i+1$ に更新して `step.1` へ戻る.

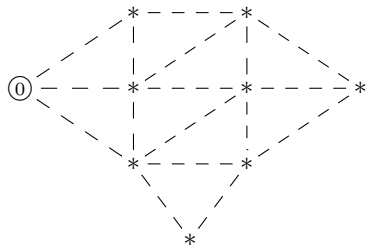
BFS アルゴリズムで得られた全域木 T を **BFS 木** と呼ぶ.

例 13-2　以下の連結グラフ $G = (V, E, \psi)$ に, 頂点 1 を始点とするような BFS アルゴリズムを適用しよう.



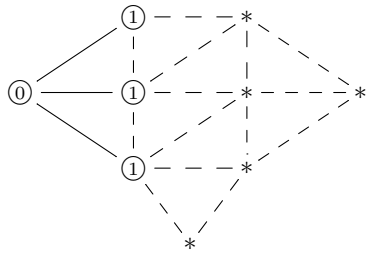
BFS アルゴリズムを適用していく流れを示そう．ここでは, 各頂点のラベルを表示する.

step.0： 任意の $v \in V$ に対して, $d(v) = *$ とラベルをつける. その後, $d(s) = 0$, $i = 0$, $E(T) = \varnothing$ とする.



step.1： 頂点 $2, 5, 7$ を探索する.

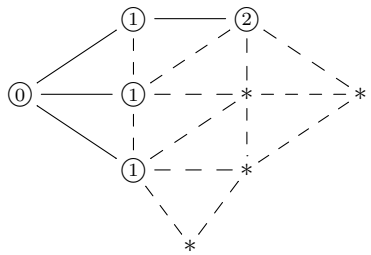
step.2： 頂点 $2, 5, 7$ のラベルを 1 として, $E(T) = \{e_1, e_2, e_3\}$ に更新して step.3 へ.



step.3： $i = 1$ として step.1 に戻る.

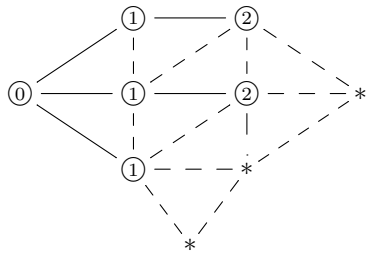
step.1： 頂点 5 に対して, 頂点 6 を探索する.

step.2： 頂点 6 のラベルを 2 として, $E(T) = \{e_1, e_2, e_3, e_6\}$ に更新する.



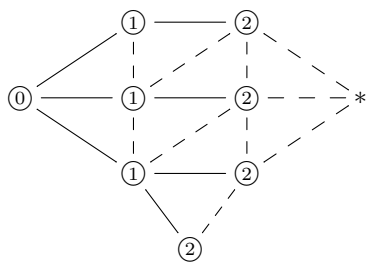
step.1： 頂点 2 に対して, 頂点 3 を探索する.

step.2： 頂点 3 のラベルを 2 として, $E(T) = \{e_1, e_2, e_3, e_5, e_6\}$ に更新する.



step.1： 頂点 7 に対して, 頂点 $8, 9$ を探索する.

step.2： 頂点 8 のラベルを 2 として, $E(T) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_{12}\}$ に更新する.

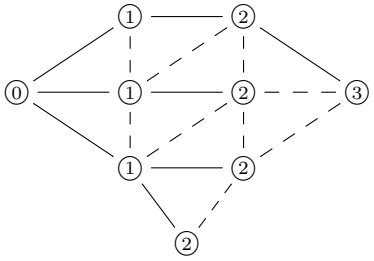


担当：茨城大学 情報科学領域　宮本 賢伍

step.3： $i = 3$ として step.1 に戻る.

step.1： 頂点 6 に対して, 頂点 4 を探索する.

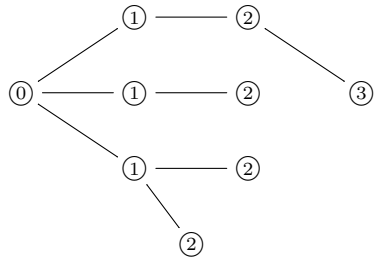
step.2： 頂点 8 のラベルを 2 として, $E(T) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_{12}\}$ に更新する.



step.1： $*$ とラベル付けられている頂点がないので, T を出力して終了.

以上で出力結果は

$T =$

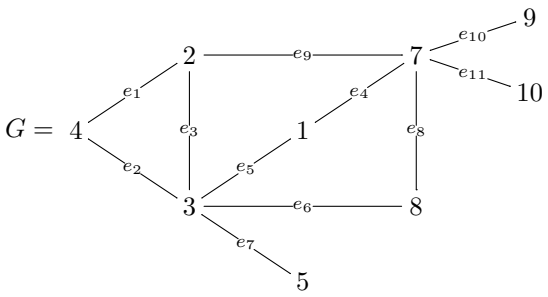


である.

注意. BFS アルゴリズムは, 非連結なグラフに対しても実行できる. その実行結果には $*$ とラベル付けられているような頂点が残るため, BFS アルゴリズムはグラフが連結かどうかを判定するアルゴリズムでもある.

レポート 13-3　次のグラフ $G = (V, E, \psi)$ に対して, 頂点 1 を始点として BFS アルゴリズムを適用することで BFS 木をひとつ出力せよ.

$G =$



担当：茨城大学 情報科学領域　宮本 賢伍