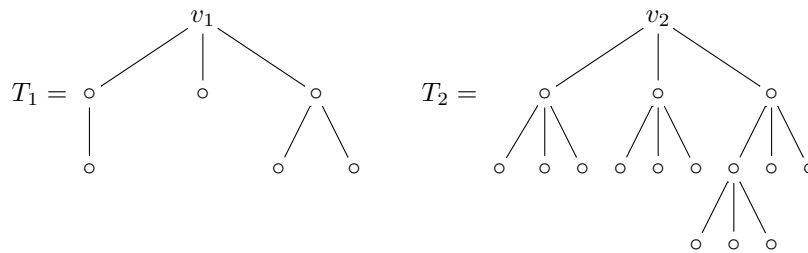


13 グラフ理論 (3) – 正則 m 分木, 幅優先探索, 深さ優先探索 –• 13-1 : 正則 m 分木

定義 13.1. 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 13.2. 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.

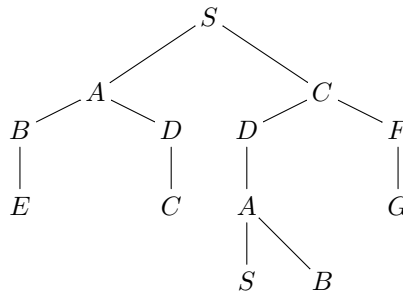
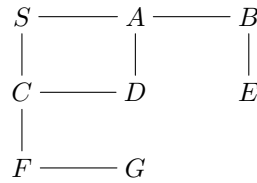
証明. 正則 m 分木 (T, v) を次のように捉えよう. 1 回の試合で m 人の人が勝負をして, そのうちの 1 人だけが勝ち上がるトーナメント戦を想定して, 優勝する 1 人を決める. つまり, 1 試合で $m - 1$ 人の人が敗北する. このとき, 葉ではない頂点 k の数は, このトーナメントの全試合数に相当する. すると, k 回の試合で, $(m - 1)k$ 人の人が敗北し, これは優勝者以外の人の数である. 試合に参加する人の数は葉の数と n 等しいから,

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

が成り立つ. □

• 13-3 : 探索木

与えられたグラフの開始頂点 S から目的頂点 G への順路を求める問題を考えよう. 開始頂点 S から調べ始めて, S と接続している頂点を順番に辿る. このとき, S を根として, それに接続する頂点が子であるような根付き木が構成される. 木は再帰的な構造をしているから, それらの子から, それを根とする下位の頂点を順に辿ることで, S を根とする木はどんどん成長する. こうして構成した木をグラフの **探索木** という. 探索木では, 同じ頂点が現れても区別するので, グラフに閉路があれば木は無限に成長し続ける. 例えば, 下図の左側のグラフの探索木の例が右側の木である.



● 13-4 : 幅優先探索と深さ優先探索

グラフとして整理されたデータを探索するとき、探索木の頂点を順に調べていくとする。どのような順序で探索を先に進めるかにより、探索木の形が変わるが、主な探索の仕方が幅優先探索と深さ優先探索である。

探索において、ある頂 v がゴールであるかを判定し、ゴールでない場合は更に展開してその子となる頂点を得ることを、 v の **調査** という。探索において、未調査の頂点のリストを **OpenList** とし、調査済みの頂点のリストを **ClosedList** と名付け、それぞれを $\text{Open}(\quad)$ 、 $\text{Closed}(\quad)$ と表す。OpenList に属する頂点は、先頭から順に調査し、調査が終了した頂点を ClosedList に移動させる。探索を進めれば、これらのリストは順に更新されていくことに注意しよう。また、頂点 A が頂点 B を展開して得られた頂点であれば、この関係を $A[B]$ と表すことにする。

例 13-2 ある探索によって、リスト

$$\text{Open}(B[S], D[S], C[A]), \quad \text{Closed}(S[\quad], A[S])$$

が得られたとする。これは、未調査の頂点が B, D, C 順に入っていて、 B, D は頂点 S を展開して得られ、 C は A から得られることを表す。また、調査済みの頂点は S, A の 2 つであり、 A の親が S 、 S は開始頂点であることを意味している。

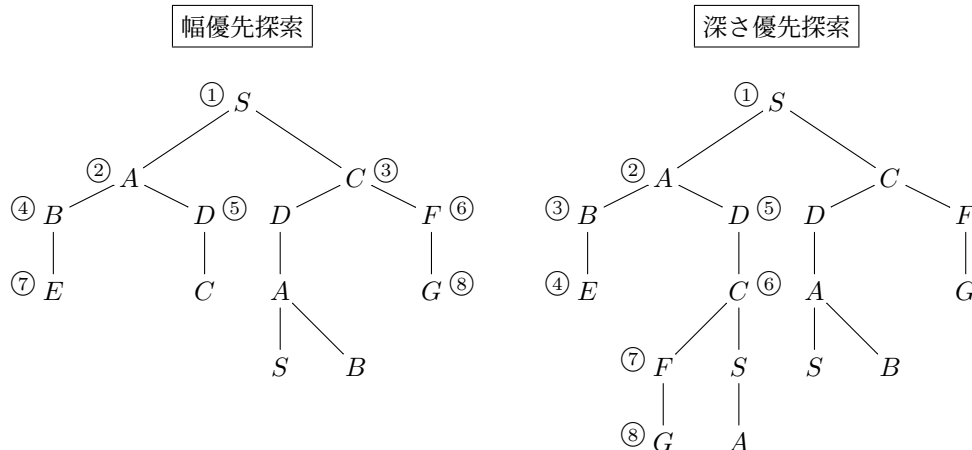
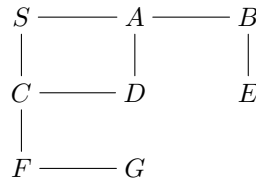
基本的な探索である幅優先探索と深さ優先探索について、その探索の仕方をまとめておこう。

用語	探索の方法
幅優先探索	到達した頂点から、到達可能な全ての隣接した頂点を探索し、その後、次の深さに潜って探索する方法。 OpenList の頂点は、調査によって得られた順に並べておく。 同じ頂点が多ければ、先に得られた方を優先する。
深さ優先探索	到達した頂点から、可能な限り深さについて深く進んだ後に戻り、他の経路を探索する方法。 OpenList の頂点は、調査によって得られたものを先頭におく。 同じ頂点が多ければ、後から得られた方を優先する。

簡単に言えば、幅優先探索は 1 段ずつ調べ、深さ優先探索は行けるところまで調べ戻ってくる探索の仕方である。それぞれの探索には得手、不得手がある。

- **幅優先探索**は、探索の際に階層ごとに進むため、最短経路を求める問題や段階的な影響伝播のシミュレーションなどに適している。一方で、必要とするメモリ量が大きくなる傾向がある。
- **深さ優先探索**は、主に再帰またはスタックを用いて実装され、木構造やグラフ構造における全経路の探索や構造解析などに用いられる。特定の解に早く到達する可能性があるが、最短経路を保証しない。

例 13-3 次のグラフの探索木に対して, S から G へのルートを幅優先探索と深さ優先探索を実行したときの探索順を示そう.



● 13-3 : 幅優先探索アルゴリズム

アルゴリズム 13.3 (幅優先探索アルゴリズム). 開始頂点を S とするような幅優先探索アルゴリズムの概略は以下の通りである.

- step.0 : $\text{Open}(S)$, $\text{Closed}(\)$ と初期化する. その後, step.2 へ.
- step.1 : OpenList が空なら, S からゴールへ至る経路は存在せず, 終了.
 そうでないなら, step.2 へ.
- step.2 : OpenList の先頭にある A を取り出し, OpenList から除く. A がゴールなら, S からゴールに至る頂点の系列を出力して終了.
 そうでなければ, A を展開して到達可能な頂点の集合 V_A を求めて step.3 へ.
 いずれの場合も, A を ClosedList の最後にいれる.
- step.3 : $V_A = \emptyset$ なら, step.1 へ戻る.
 $V_A \neq \emptyset$ なら, V_A に属する頂点 B に対して, $B[A]$ が **OpenList もしくは ClosedList に存在していなければ**, OpenList の**末尾**に追加して step.1 に戻る.

● 13-4 : 深さ優先探索アルゴリズム

アルゴリズム 13.4 (深さ優先探索アルゴリズム). 開始頂点を S とするような深さ優先探索アルゴリズムの概略は以下の通りである.

- step.0 : $\text{Open}(S)$, $\text{Closed}(\)$ と初期化する. その後, step.2 へ.
- step.1 : OpenList が空なら, S からゴールへ至る経路は存在せず, 終了.
 そうでないなら, step.2 へ.
- step.2 : OpenList の先頭にある A を取り出し, OpenList から除く. A がゴールなら, S からゴールに至る頂点の系列を出力して終了.
 そうでなければ, A を展開して到達可能な頂点の集合 V_A を求めて step.3 へ.
 いずれの場合も, A を ClosedList の最後にいれる.

step.3: $V_A = \emptyset$ なら, step.1 へ戻る.

$V_A \neq \emptyset$ なら, V_A に属する頂点 B に対して, $B[A]$ が **ClosedList** に存在していなければ, **OpenList** の先頭に追加して step.1 に戻る.

小テスト 13-1 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.

小テスト 13-2 次のアルゴリズムについて, 正しい記述であれば ○, 間違っていれば × を選べ.

アルゴリズム (Input) 連結なグラフ $G = (V, E, \psi)$ と始点 $S \in V$.

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, $d(S) = 0, i = 0, E(T) = \emptyset$ 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

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

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.

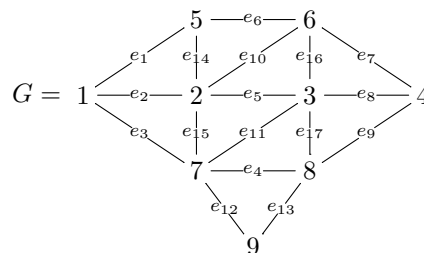
- (1) このアルゴリズムは幅優先探索をするアルゴリズムである.
- (2) このアルゴリズムを連結なグラフ G に適用すれば, G の全域木が得られる.
- (3) このアルゴリズムによって, グラフ G が連結であるかがわかる.
- (4) このアルゴリズムは連結なグラフにしか適用できず, その結果, アルゴリズム終了時には $*$ とラベル付けされる頂点が存在しない.
- (5) このアルゴリズムを実行したときの $d(v)$ は, 開始頂点 S から頂点 v への道の長さである.

演習問題 13-1 $m > 1$ を自然数とする. 全ての葉の深さが h の正則 m 分木の頂点集合を V とすれば, V の濃度が

$$\frac{m^{h+1} - 1}{m - 1}$$

で与えられることを示せ.

演習問題 13-2 次のグラフ $G = (V, E, \psi)$ に対して, 頂点 1 を始点として, 演習 13-2 のアルゴリズムを適用した結果を出力せよ.



演習問題 13-3 次のグラフの 1 から 9 までのルートを幅優先探索および深さ優先探索をそれぞれ実行して, OpenList と ClosedList の変化を示せ. この探索による探索木もかけ. ただし, 展開して得られた頂点が複数ある場合は, 数が大きいほうが左に配置されるものとする.

