組合せとグラフの理論　（塩田）

— 幅優先探索、深さ優先探索 —

幅優先探索、深さ優先探索ともに、根と呼ばれる頂点から木を伸ばしていって全域木を作るアルゴリズムである。

1　幅優先探索（Breadth First Search）

現在地から 幅広く 未探索点を探してゆく探索方法。

幅優先探索アルゴリズム

入力： グラフ G と、その頂点 v

出力： v を根とする幅優先探索木 T

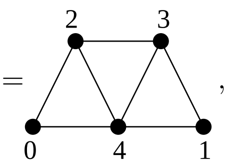
1°　v を探索済みとし、T は v 一点だけからなる木とする。キュー $Q = \{v\}$ を作る。

2°　Q の先頭から x を取り出し現在地とする。

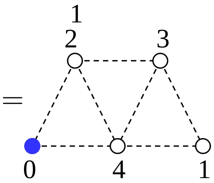
3°　現在地 x の未探索な隣接点のすべてを、番号の小さい順に $y_1, y_2, \cdots, y_s$ とする。（未探索隣接点が無いこともある。）

4°　$y_1, y_2, \cdots, y_s$ を探索済みとして Q の最後尾にこの順で追加し、T には辺 $xy_1, \cdots, xy_s$ をすべて付加する。

5°　Q が空（くう）リストになれば終了。そうでなければ 2° へ戻る。

例 1　入力を $G =$ 

，　$v = 0$ とする。

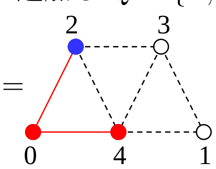
1°　0 を探索済みとし、$T =$ 

, $Q = \{0\}$ とする。（青い頂点 は Q の先頭で、次の現在地になる。）

2°　Q の先頭から $x = 0$ を取り出して現在地とし、$Q = \emptyset$.

3°　現在地 $x = 0$ の未探索な隣接点は $y_1 = 2,\ y_2 = 4$ のふたつ。

4°　$y_1 = 2,\ y_2 = 4$ を探索済みとして Q の最後尾に追加し $Q = \{2, 4\}$,

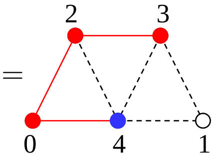
T には辺 $xy_1 = 02$ と $xy_2 = 04$ を付加して $T =$ 

とする。

2°　Q の先頭から $x = 2$ を取り出して現在地とし、$Q = \{4\}$.

3° 現在地 $x=2$ の未探索な隣接点は $y_1=3$.

4° $y_1=3$ を探索済みとして Q の最後尾に追加し $Q=\{4,3\}$,

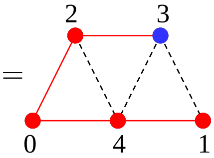
T には辺 $xy_1=23$ を付加して $T=$ 

とする。

2° Q の先頭から $x=4$ を取り出して現在地とし、$Q=\{3\}$.

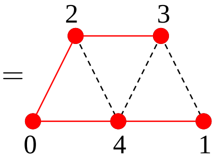
3° 現在地 $x=4$ の未探索な隣接点は $y_1=1$.

4° $y_1=1$ を探索済みとして Q の最後尾に追加し $Q=\{3,1\}$,

T には辺 $xy_1=41$ を付加して $T=$ 

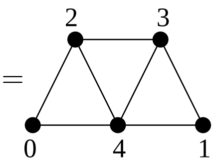
とする。

手でやるときは、未探索点が無くなったのでここで終了すればよく、

$T=$ 

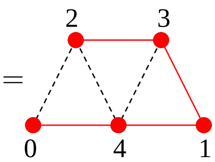
となる。(アルゴリズム的には、このあと Q から 3, 1 が取り出されて $Q=\emptyset$ となって終了する。) 手順を表にしてまとめておくと

現在地 x	x の未探索な隣接点	T に付加する辺	Q
			$\{0\}$
0	2, 4	02, 04	$\{2,4\}$
2	3	23	$\{4,3\}$
4	1	41	$\{3,1\}$

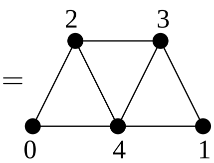
例 2 入力を同じグラフ $G=$ 

とし、根を $v=1$ とすると、手順は

現在地 x	x の未探索な隣接点	T に付加する辺	Q
			$\{1\}$
1	3, 4	13, 14	$\{3,4\}$
3	2	32	$\{4,2\}$
4	0	40	$\{2,0\}$

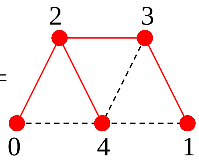
となり、幅優先探索木は $T=$ 

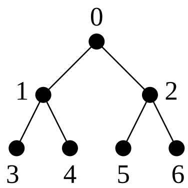
となる。

例 3 三たび同じグラフ $G=$ 

で、根を $v=2$ とした場合、手順は

現在地 x	x の未探索な隣接点	T に付加する辺	Q
			$\{2\}$
2	0, 3, 4	20, 23, 24	$\{0, 3, 4\}$
0	無し	無し	$\{3, 4\}$
3	1	31	$\{4, 1\}$

となり、幅優先探索木は $T =$  となる。

例 4　二分木 $T =$  はそれ自体が全域木であるが、$v = 0$ を根とする幅優先探索の手順は

現在地 x	x の未探索な隣接点	探索木に付加する辺	Q
			$\{0\}$
0	1, 2	01, 02	$\{1, 2\}$
1	3, 4	13, 14	$\{2, 3, 4\}$
2	5, 6	25, 26	$\{3, 4, 5, 6\}$

となる。すなわち頂点が探索される順番は

$$0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6$$

になる。(後ろの例 8 と比較せよ。)

2　深さ優先探索 (Depth First Search)

現在地に隣接する未探索点がある限り先へ先へ探索を進め、隣接点が全て探索済みになったらバックトラックを掛ける探索方法。

深さ優先探索アルゴリズム

入力： グラフ G とその頂点 v

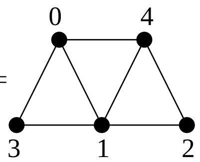
出力： v を根とする G の深さ優先探索木 T

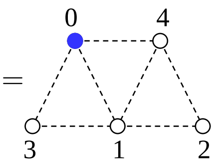
1°　v を探索済みとし、T は v 一点だけからなる木 とする。最初の現在地を $x = v$ とする。

2°　現在地 x の未探索な隣接点があれば、そのうち番号の一番小さいものを y とする。そうでなければ 4° へ。

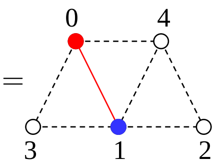
3°　y を探索済みとし、T に辺 xy を付加する。次の現在地を (新 x) $= y$ として 2° へ戻る。(x を「y の親」と呼ぶ。)

4°　$x = v$ であれば終了。そうでなければ「 x の親」を次の現在地 (新 x) として 2° へ戻る。(バックトラック)

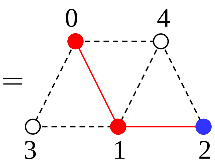
例 5　入力を $G =$  , $v = 0$ とする。(例 1-3 の G とは頂点番号を変えてあるので注意。)

1°　0 を探索済みとし、$T =$  とする。$x = 0$ を最初の現在地とする。　(青い頂点が現在地。)

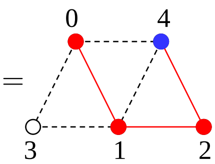
2°　現在地 $x = 0$ の未探索な隣接点 1, 3, 4 のうち番号が一番小さいものは $y = 1$.

3°　$y = 1$ を探索済みとし、T には辺 01 を付加して $T =$  , (新 x) $= 1$.

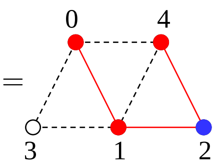
2°　現在地 $x = 1$ の未探索な隣接点 2, 3, 4 のうち番号が一番小さいものは $y = 2$.

3°　$y = 2$ を探索済みとし、T には辺 12 を付加して $T =$  , (新 x) $= 2$.

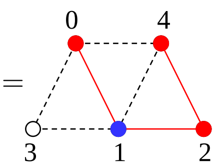
2°　現在地 $x = 2$ の未探索な隣接点は $y = 4$.

3°　$y = 4$ を探索済みとし、T には辺 24 を付加して $T =$  , (新 x) $= 4$.

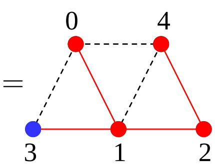
2°　現在地 $x = 4$ には未探索な隣接点は無い.

4°　4 の親 2 へバックトラックし、$T =$  , (新 x) $= 2$.

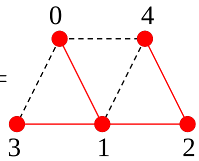
2°　現在地 $x = 2$ にも未探索な隣接点は無い.

4°　2 の親 1 へバックトラックし、$T =$  , 新 $x = 1$.

2°　現在地 $x = 1$ の未探索な隣接点は $y = 3$.

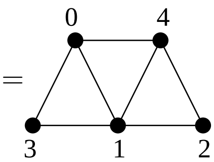
3°　$y = 3$ を探索済みとし、T には辺 13 を付加して $T =$  , (新 x) $= 3$.

手でやるときは、未探索点が無くなったのでここで終了すればよく、

$T =$ 

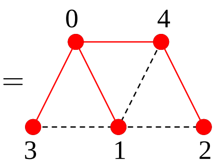
となる。（アルゴリズム的にはこのあと、1, 0 へバックトラックして終了となる。）手順を表にしてまとめておくと

現在地	未探索な隣接点	動作	T に付加する辺
0	1, 3, 4	1 へ進む	01
1	2, 3, 4	2 へ進む	12
2	4	4 へ進む	24
4	無し	2 へバックトラック	無し
2	無し	1 へバックトラック	無し
1	3	3 へ進む	13

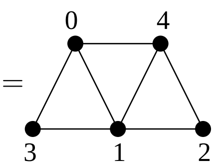
例 6　入力を同じグラフ $G =$ 

とし、根を $v = 1$ とすると、手順は

現在地	未探索な隣接点	動作	T に付加する辺
1	0, 2, 3, 4	0 へ進む	10
0	3, 4	3 へ進む	03
3	無し	0 へバックトラック	無し
0	4	4 へ進む	04
4	2	2 へ進む	42

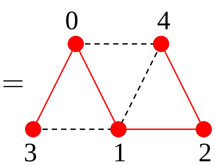
となり、深さ優先探索木は $T =$ 

となる。

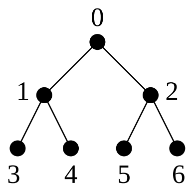
例 7　三たび同じグラフ $G =$ 

で、根を $v = 3$ とした場合、手順は

現在地	未探索な隣接点	動作	T に付加する辺
3	0, 1	0 へ進む	30
0	1, 4	1 へ進む	01
1	2, 4	2 へ進む	12
2	4	4 へ進む	24

となり、深さ優先探索木は $T =$ 

となる。

例 8　二分木 $T=$ 

で $v=0$ を根とする深さ優先探索の手順は

現在地	未探索な隣接点	動作	探索木に付加する辺
0	1, 2	1 へ進む	01
1	3, 4	3 へ進む	13
3	無し	1 へバックトラック	無し
1	4	4 へ進む	14
4	無し	1 へバックトラック	無し
1	無し	0 へバックトラック	無し
0	2	2 へ進む	02
2	5, 6	5 へ進む	25
5	無し	2 へバックトラック	無し
2	6	6 へ進む	26

となる。すなわち頂点が探索される順番は

$$0,\ 1,\ 3,\ 4,\ 2,\ 5,\ 6$$

になる。

幅優先探索がキューを用いて記述されたのに対し、深さ優先探索も次のようにスタックを用いて記述できる：

深さ優先探索アルゴリズム（スタックを用いた記述）

入力： グラフ G と頂点 v

出力： v を根とする G の深さ優先探索木 T

1°　v を探索済みとし、T は v 一点だけからなる木 とする。空スタック $S=\{\ \}$ を作る。

2°　現在地を $x=v$ とする。

3°　現在地 x に未探索な隣接点があれば、そのうち番号が一番小さいものを y とする。そうでなければ 5° へ。

4°　y を探索済みとして T には辺 xy を付加する。S の先頭に x を追加し、次の現在地を (新 x) $=y$ として 3° へ。

5°　S が空であれば終了。そうでなければ S の先頭から x を取り出して現在地とし 3° へ。(バックトラック)