８. 任意のデータ構造（グラフの表現とアルゴリズム）

- ８ー１. グラフの数学的定義
  - 集合によるグラフ定義
  - グラフの図式表現
- ８ー２. 配列でのグラフ表現
  - 隣接行列
  - 接続行列
- ８ー３. 連結リストによるグラフ表現
  - 隣接リスト
- ８ー４. グラフ上のアルゴリズム

1

８ー１. グラフの数学的表現

- n個の頂点の集合を

$$V = \{v_1, v_2, \cdots, v_n\}$$

とし、

- m個の辺の集合を

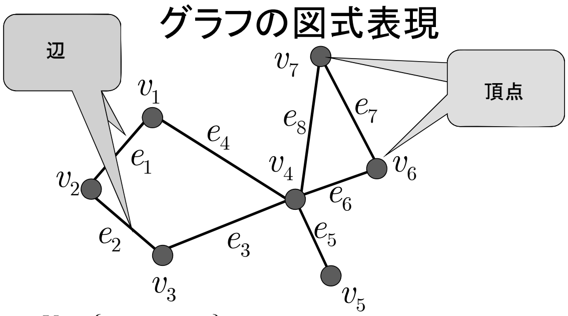
$$E = \{e_1, e_2, \cdots, e_m\}$$

とする。ただし、辺 $e_i \in E$ は、
頂点の対、すなわち $e_i = (v_p, v_q) \in V \times V$ と表せる。
よって、 $E \subseteq V \times V$ である。

- このとき、グラフG は頂点集合V と辺集合E の対 $G = (V, E)$で表される。

2

グラフの図式表現



$V = \{v_1, v_2, \cdots, v_7\}$

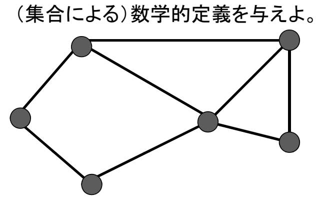
$E = \{e_1, e_2, \cdots, e_8\}$
$= \{(v_1, v_2), (v_2, v_3), \cdots, (v_7, v_4)\}$

$G = (V, E)$

3

練習

（1） 図式表現で与えられた次のグラフの（集合による）数学的定義を与えよ。



（2） 次の数学的表現で与えられたグラフの図式表現を求めよ。

$G = (V, E)$

$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$
$= \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_6),$
$(v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_6)\}$

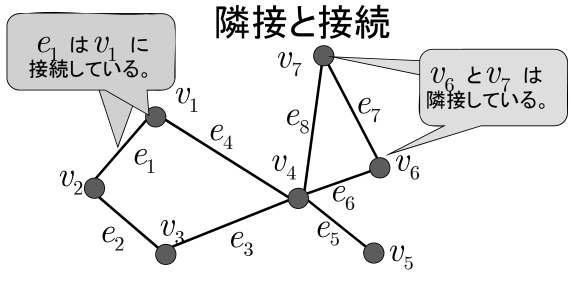
4

８ー２. 配列でのグラフ表現

- 配列（行列）でグラフを表現する方法には主に2つの方法がある。
  - 隣接行列　点同士の隣接関係に注目する方法
  - 接続行列　点と辺の接続関係に注目する方法。

5

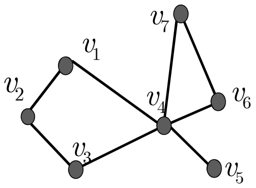
隣接と接続



点 v_iと点v_j に対して、$(v_i, v_j) \in E$ であるとき、その2点は隣接している。
点 v_i と辺e_j に対して、$e_j = (v_i, v_k)$ であるとき、辺 e_j は点 v_i に接続している。

6

隣接行列



	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	1	0	0	0
v_2	1	0	1	0	0	0	0
v_3	0	1	0	1	0	0	0
v_v	1	0	1	0	1	1	1
v_5	0	0	0	1	0	0	0
v_6	0	0	0	1	0	0	1
v_7	0	0	0	1	0	1	0

無向グラフの隣接行列は、対称行列。

7

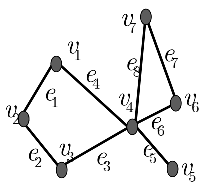
隣接行列の性能

n:頂点数$|V|$ 、
m:辺数 $|E|$ とする。

記憶量	$O(n^2)$ 領域
2点間の隣接関係チェック	$O(1)$ 時間
1点からの隣接関係チェック	$O(n)$ 時間

8

接続行列



	e_1	e_2	e_3	e_4	e_5	e_6	e_8	e_9
v_1	1	0	0	1	0	0	0	0
v_2	1	1	0	0	0	0	0	0
v_3	0	1	1	0	0	0	0	0
v_4	0	0	1	1	1	1	0	1
v_5	0	0	0	0	1	0	0	0
v_6	0	0	0	0	0	1	1	0
v_7	0	0	0	0	0	0	1	1

9

接続行列の性能

n:頂点数$|V|$ 、
m:辺数 $|E|$ とする。

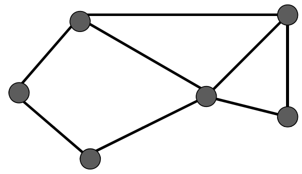
記憶量	$O(nm)$ 領域
接続関係のチェック	$O(1)$ 時間
1点からの隣接関係チェック	$O(m)$ 時間

$m = O(n^2)$ にも注意する。

10

練習

（1） 図式表現で与えられた次のグラフの隣接行列およびせ接続行列を与えよ。



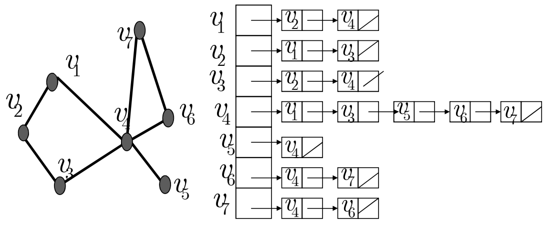
11

8－3. 隣接リスト表現

- 隣接行列では、隣接関係にないものまで保存していた。
- この無駄をリスト構造によって、解消する。

12

隣接リスト



13

隣接リストの性能

n:頂点数$|V|$ 、
m:辺数$|E|$ とする。

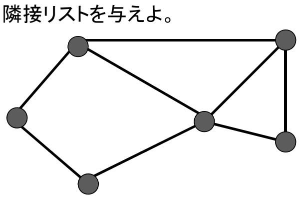
記憶量	$O(n+m)$ 領域
２点間の隣接関係のチェック	$O(\Delta)$ 時間
１点からの隣接関係チェック	$O(\Delta)$ 時間

Δ： 最大次数
$\Delta = O(n)$ にも注意する。

14

練習

（1） 図式表現で与えられた次のグラフの隣接リストを与えよ。



15

８－４. グラフ上のアルゴリズム

- グラフの表現法により、アルゴリズムの計算量に差が生じる。
  - →問題やその解法（アルゴリズム）にしたがって、表現法を選択する必要がある。
- グラフ上の基本的探索技法
  - 深さ優先探索とスタック
  - 幅優先探索とキュー

16

グラフ探索アルゴリズム

1. 始点sを選び、ラベルL＝1にセットする。
   ① sにラベルLをつける。
   ② sの隣接点を集合Sに入れる。
2. 集合Sから一点pを取り出す。
   ① L＋＋とし、pにラベルLをつける。
   ② pの隣接点でラベルがついていないものを、集合Sに入れる。
3. 全ての点にラベルがつくまで、2. を繰り返す。

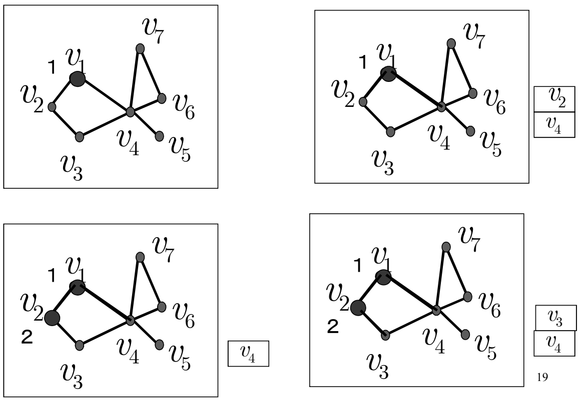
17

深さ優先探索(Depth First Search)

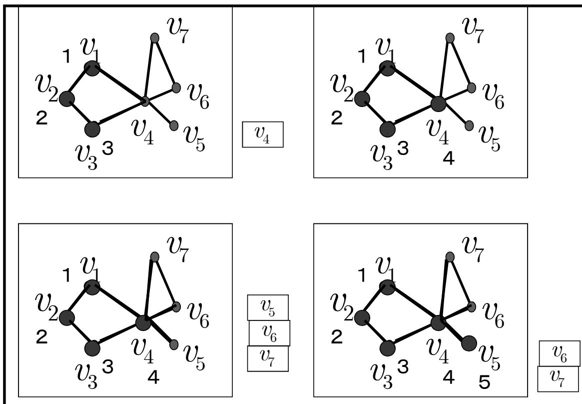
```
dfs(vertex v){
    stack s;    L=1;
    while(L<n){
        Label[v]=L;
        for(vの隣接リストlist[v]){
            w ∈ list[v] が未ラベルなら、
            s.push(w);}
        v=s.pop();
        L++:        }
}
```

18

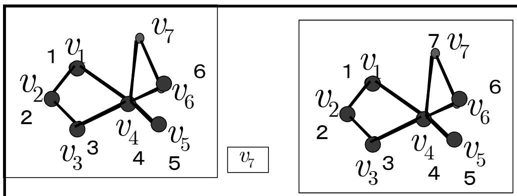
DFSの動作



19



20



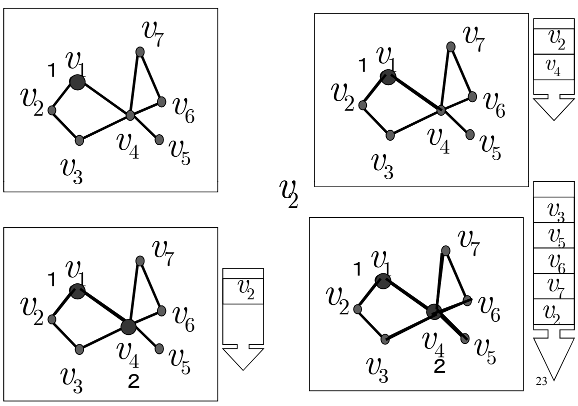
21

幅優先探索(Breadth First Search)

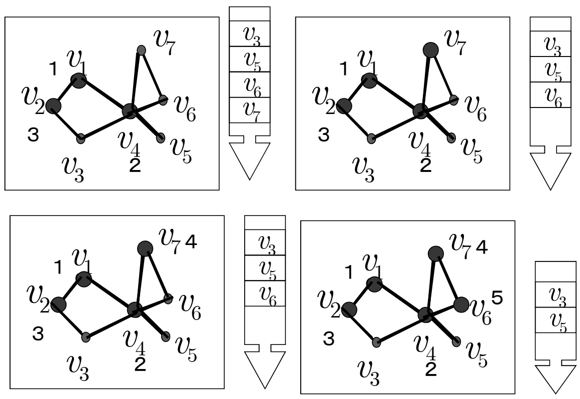
```
bfs(vertex v){
        queue q;   L=1;
        while(L<n){
                Label[v]=L;
                for(vの隣接リストlist[v]){
                        w ∈ list[v] が未ラベルなら、
                        q.enqueue(w);}
                v=q.dequeue();
                L++:          }
}
```

22

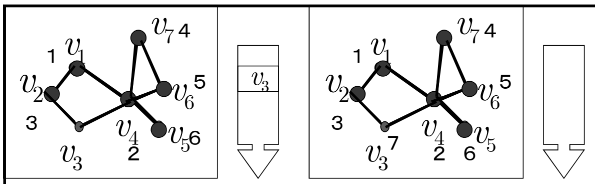
BFSの動作



23



24



25

グラフ探索アルゴリズムの性能

すべての辺をしらべながら、全ての頂点を訪れるので、
いかの計算量で実現可能である。

	時間計算量	領域計算量
深さ優先探索	$O(n+m)$	$O(n)$
幅優先探索	$O(n+m)$	$O(n)$

n:点数、m:辺数

26