



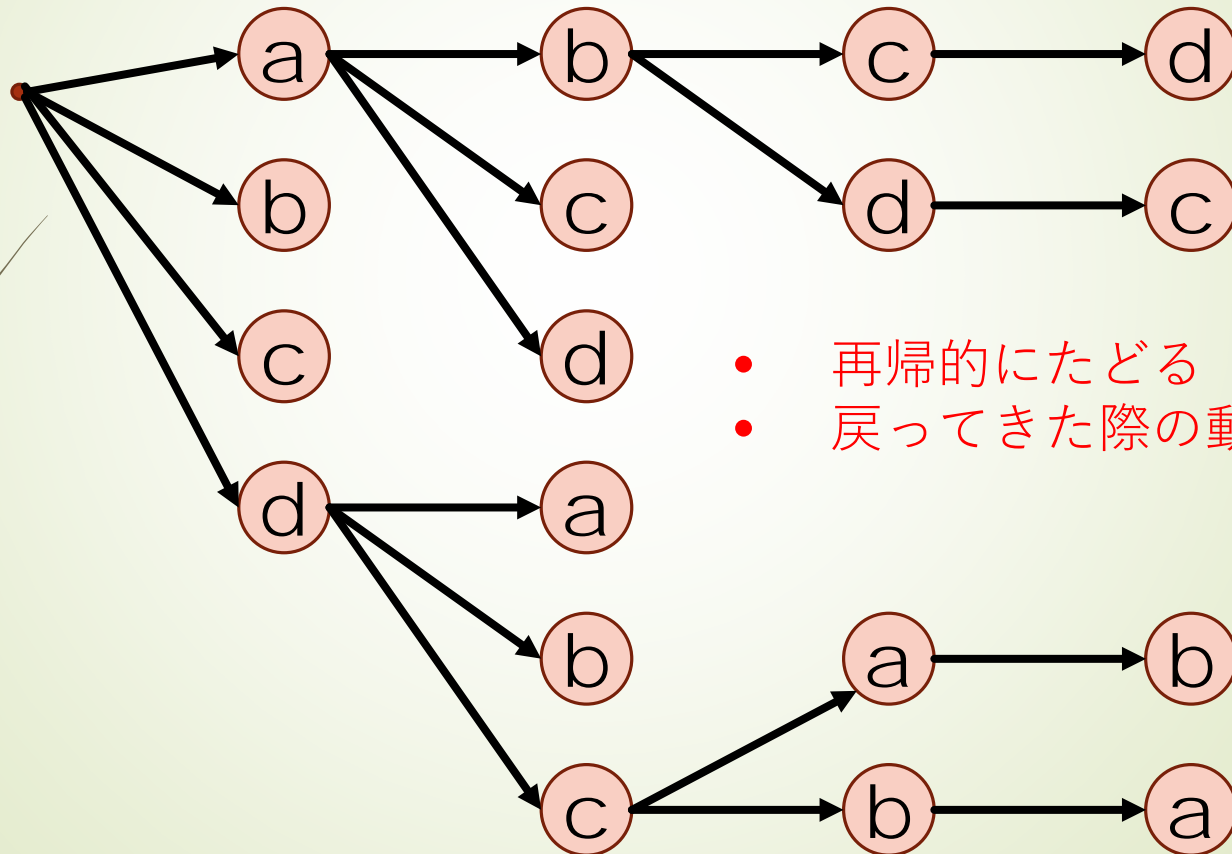
# Euler閉路と Hamilton閉路

2018年度

担当：只木進一（理工学部）

# 列挙の考え方の基本

文字{a,b,c,d}をすべて使った文字列

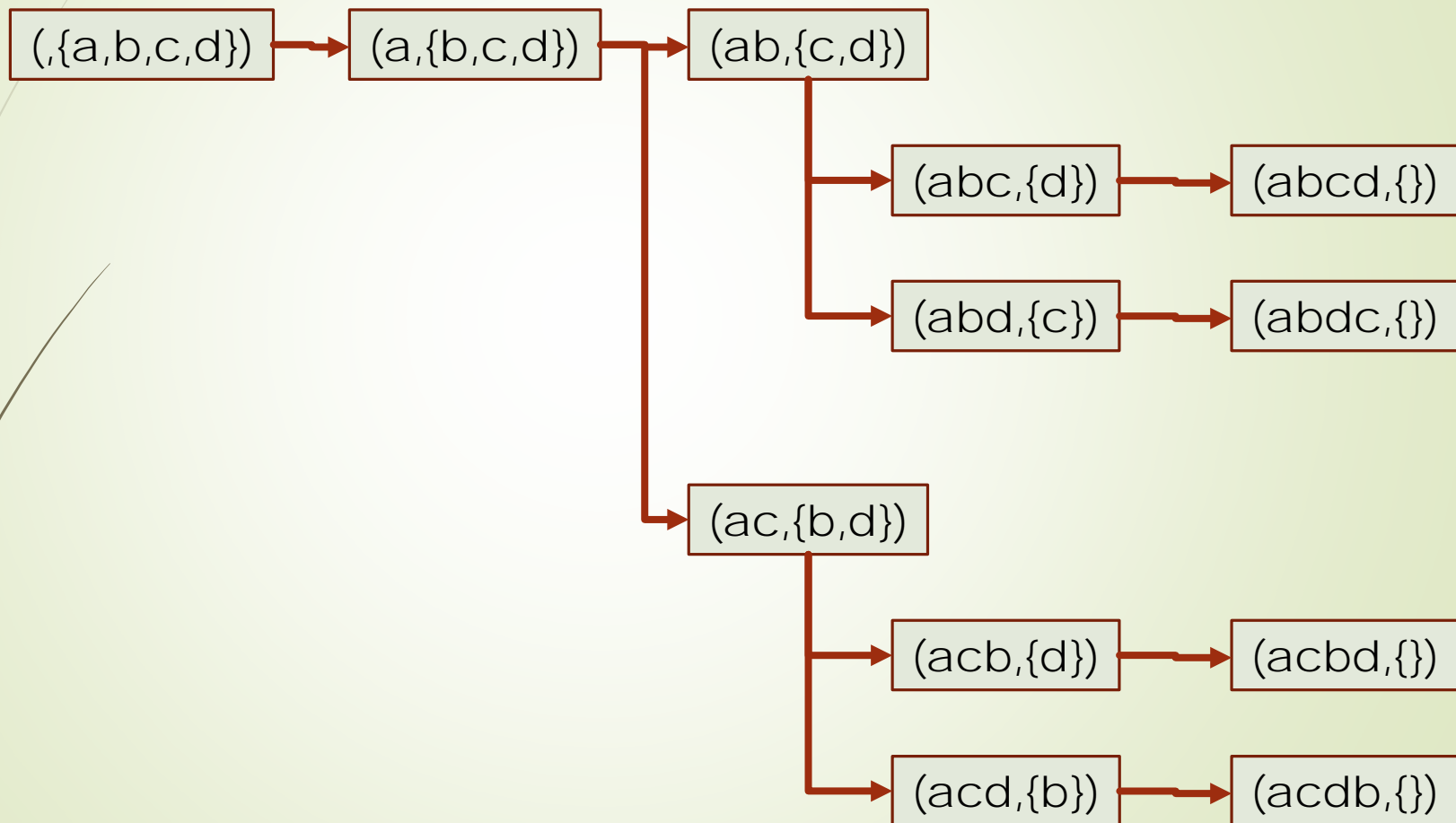


- 再帰的にたどる
- 戻ってきた際の動きに注意

- ▶ 使える文字の集合  $C$
- ▶ あるところまで使っていない文字の集合  $R \subseteq C$
- ▶ 文字列  $s$

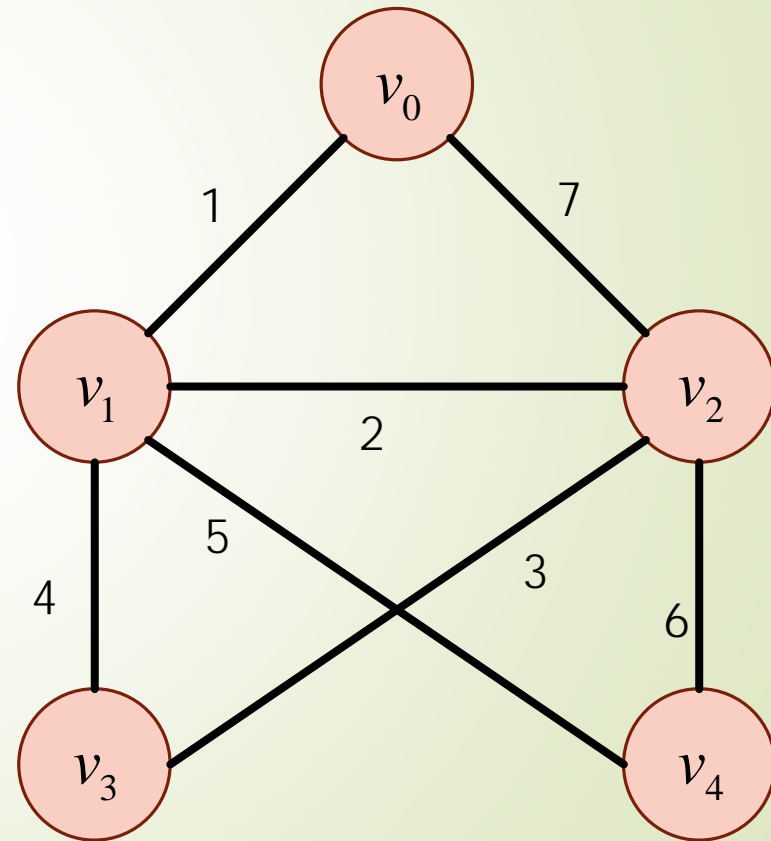
```
enumerate( $s, R$ ){  
  if ( $R == \emptyset$ ) { $s$ を出力}  
  else {  
    foreach ( $s' \in R$ ){  
       $Q = R \setminus \{s'\}$   
      enumerate( $s + s', Q$ )  
    }  
  }  
}
```

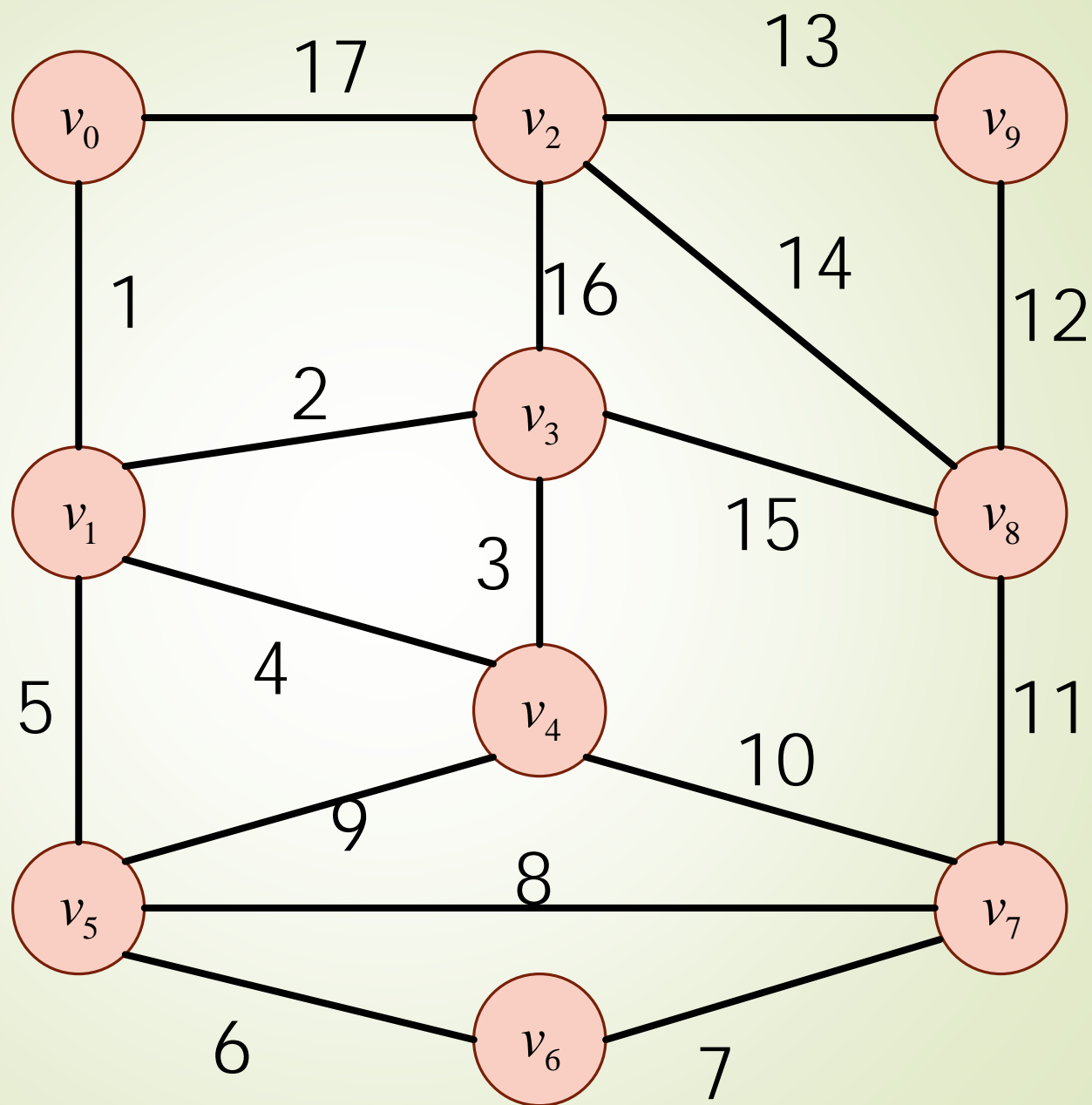
## enumerateの引数の変化



# Euler閉路(Euler Circle)

- 無向グラフが対象
- 一筆書き
  - 全ての弧を一度ずつ経由して始点に戻る道
- 全ての点の次数が偶数
  - 次数が奇数だったらどうなる？





全ての閉路を見つける方法は？

# Euler閉路の列挙の方針

## 全てのEuler閉路を見つける

- 使用した弧の列を管理する
- 深さ優先で、一つの閉路を見つける
  - 使った弧の一覧を保持
- 分岐点まで戻って、他の閉路を見つける
  - 戻るたびに、対応する弧（一つの閉路の構成要素）を一覧から削除
- 上記を繰り返す

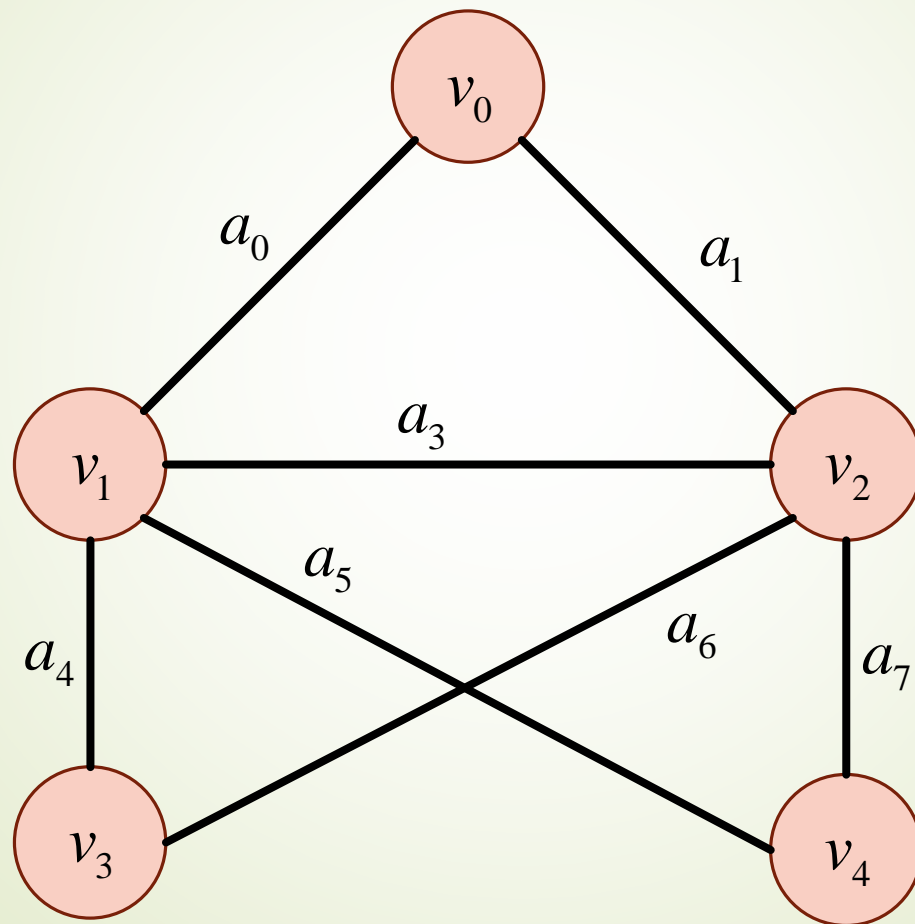
# Euler閉路の列挙アルゴリズム

```
search( $v, A_{\text{Euler}}$ ) {  
  if( ( $v == r$ )  $\wedge$  ( $|A| == |A_{\text{Euler}}|$ ) ) {  
    見つかったEuler閉路を保存  
  } else {  
    forall(  $a \in \delta v$  ) {  
      if(  $a \notin A_{\text{Euler}}$  ) {  
         $A'_{\text{Euler}} = A_{\text{Euler}} \cup \{a\}$   
         $w = \partial a \setminus \{v\}$   
        search(  $w, A'_{\text{Euler}}$  )  
      }  
    }  
  }  
}
```

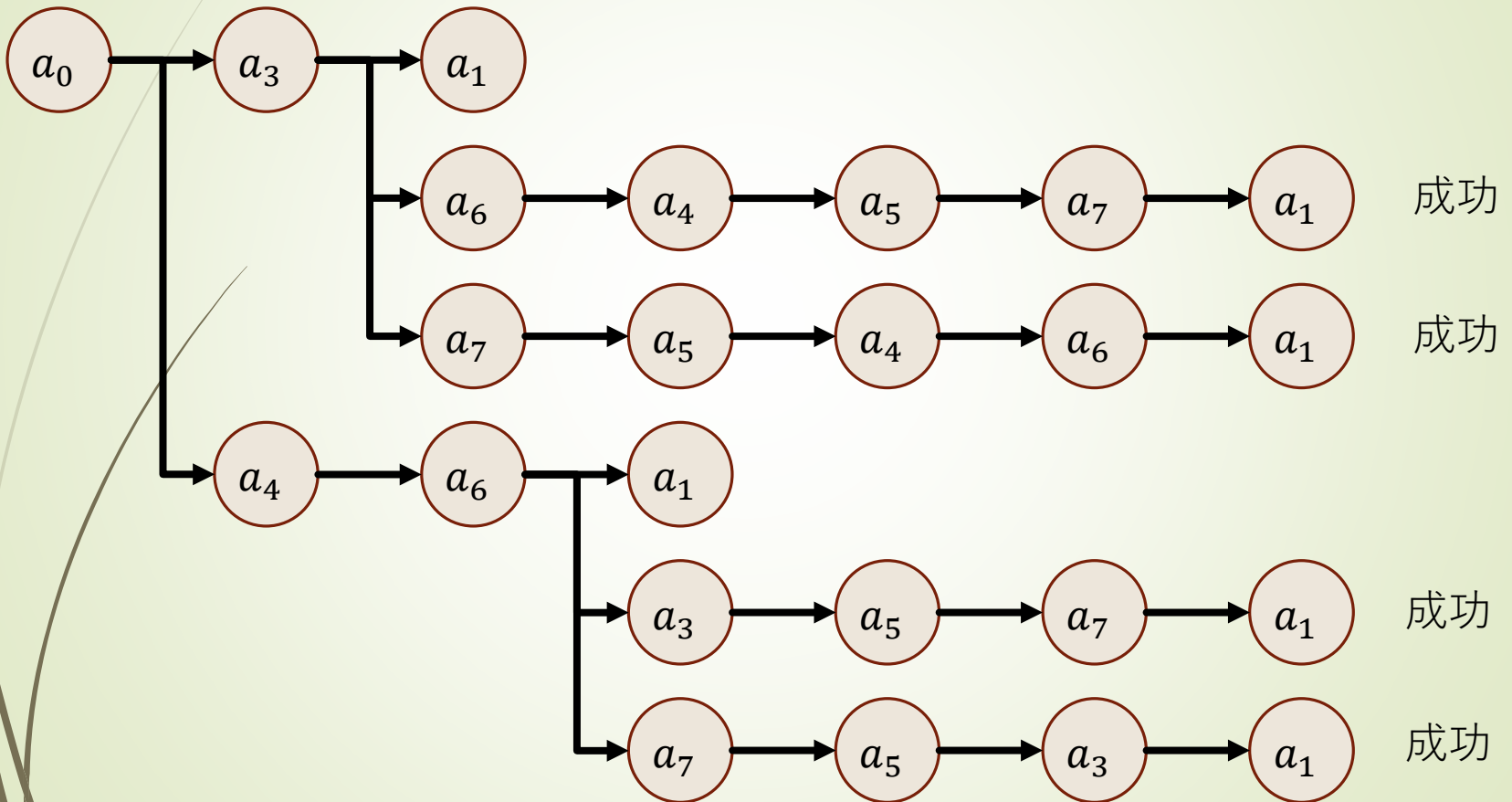
$A_{\text{Euler}}$  : 既に経由した弧の列、初期値は $\emptyset$   
 $r$  : 始点



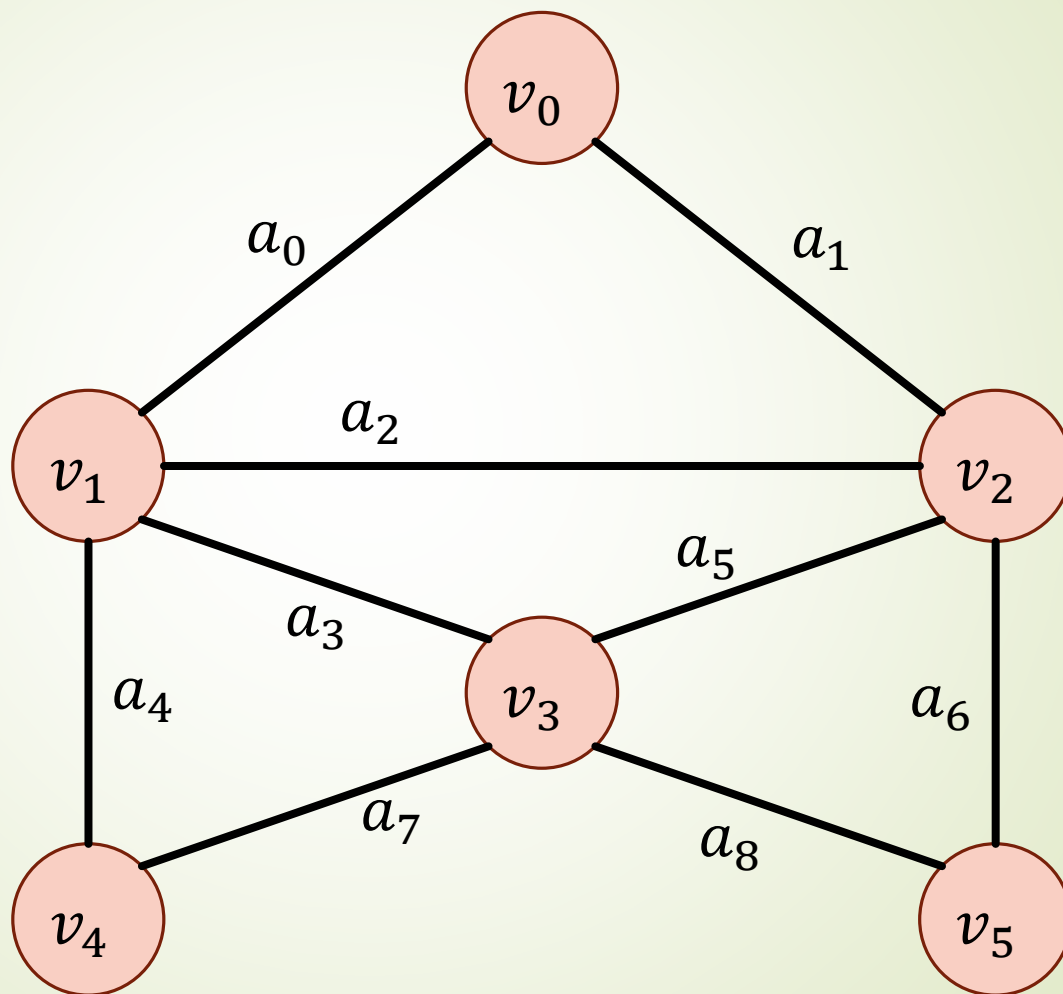
## 列挙の例

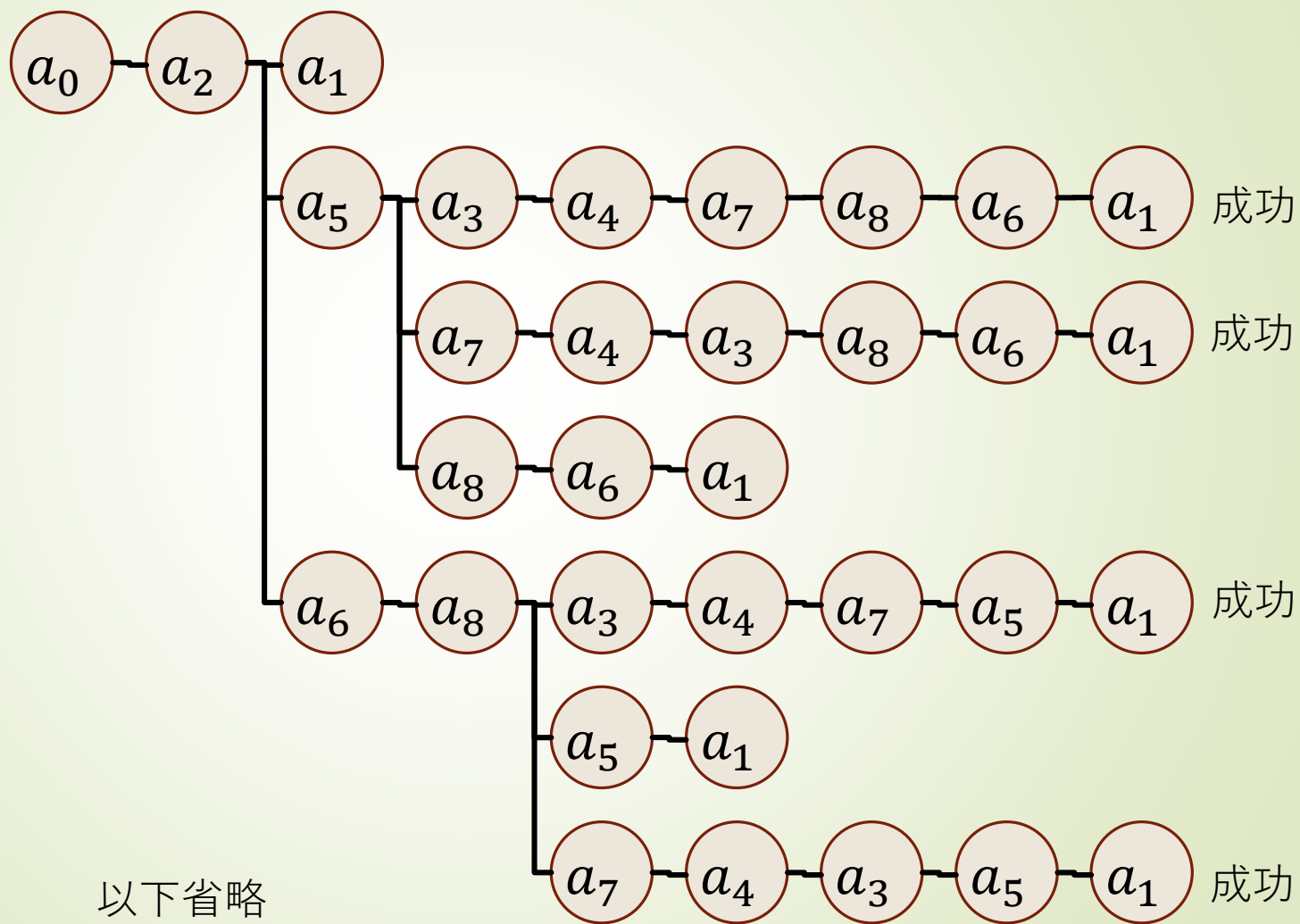


## 探索の様子



## 例2





# Hamilton 閉路

- 無向グラフが対象
- 全ての点を一度ずつ経由して始点に戻る道
- 巡回セールスマン問題の厳密解を得る際に必要
- 経由した頂点の列の管理が必要

# Hamilton閉路の列挙の方針

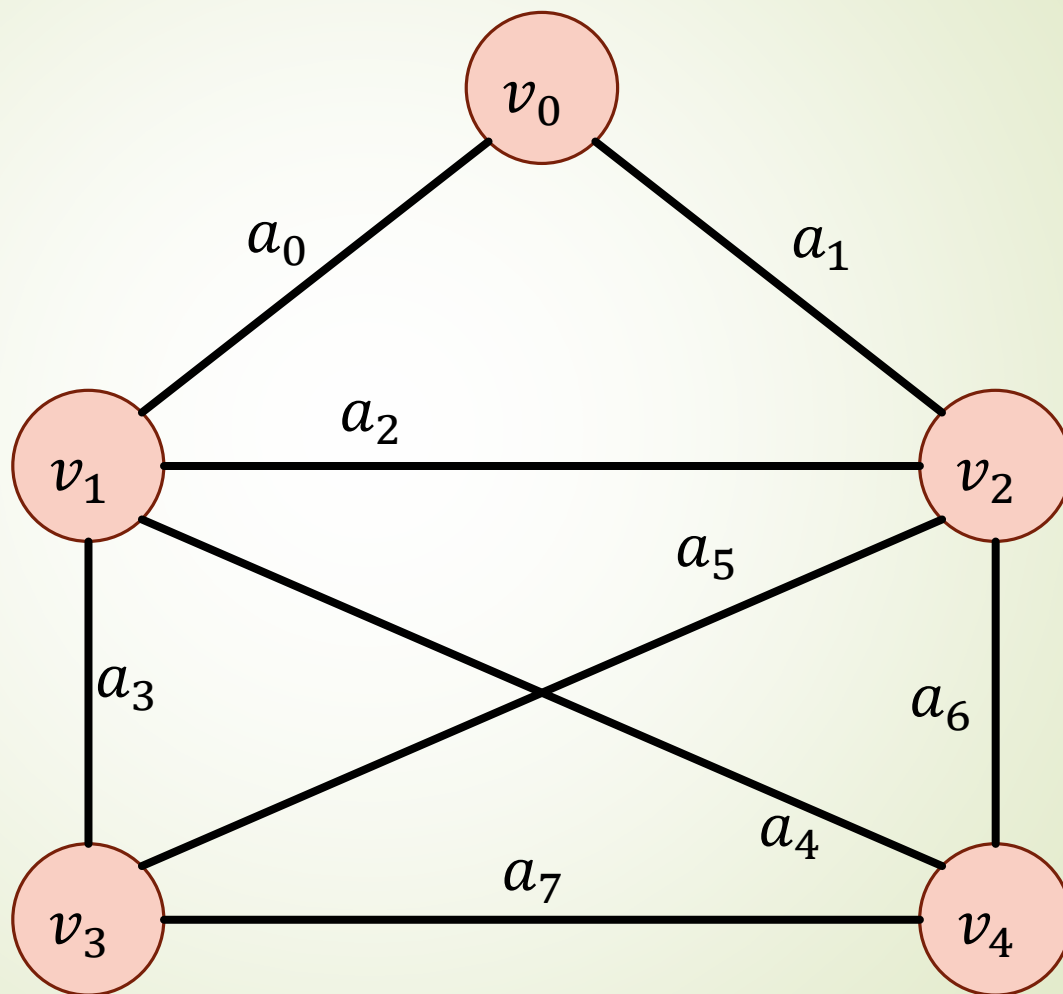
- ➡ 経由した頂点の列を管理する
- ➡ 深さ優先で、一つの閉路を見つける
  - ➡ 経由した頂点の一覧を保持
- ➡ 分岐点まで戻って、他の閉路を見つける
  - ➡ 戻るときに、対応する頂点（一つの閉路の構成要素）を一覧から削除
- ➡ 上記を繰り返す

# Hamilton閉路の列挙アルゴリズム

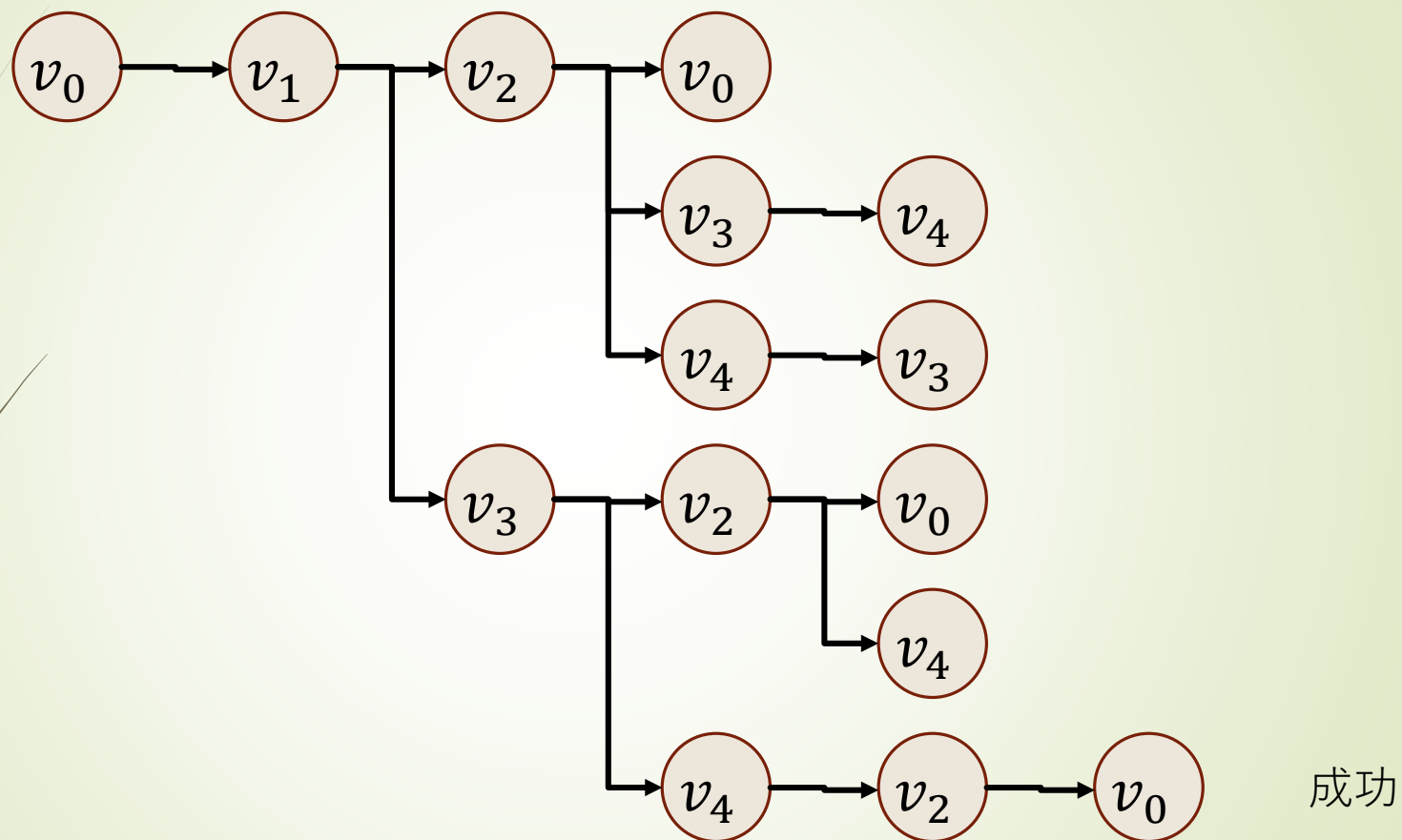
```
search(  $v, L$  ){  
  forall(  $a \in \delta v$  ) {  
     $w = \partial a \setminus \{v\}$   
    if( ( $w == r$ )  $\wedge$  ( $|L| == |V|$ ) ) {見つけた閉路を保存  
  } else {  
    if (  $w \notin L$  ) {  
       $L' = L \cup \{w\}$   
      search( $w, L'$ )  
    }  
  }  
}
```

$L$ : 経由した点のリスト、初期値は  
 $[r]$   
 $r$ : 始点

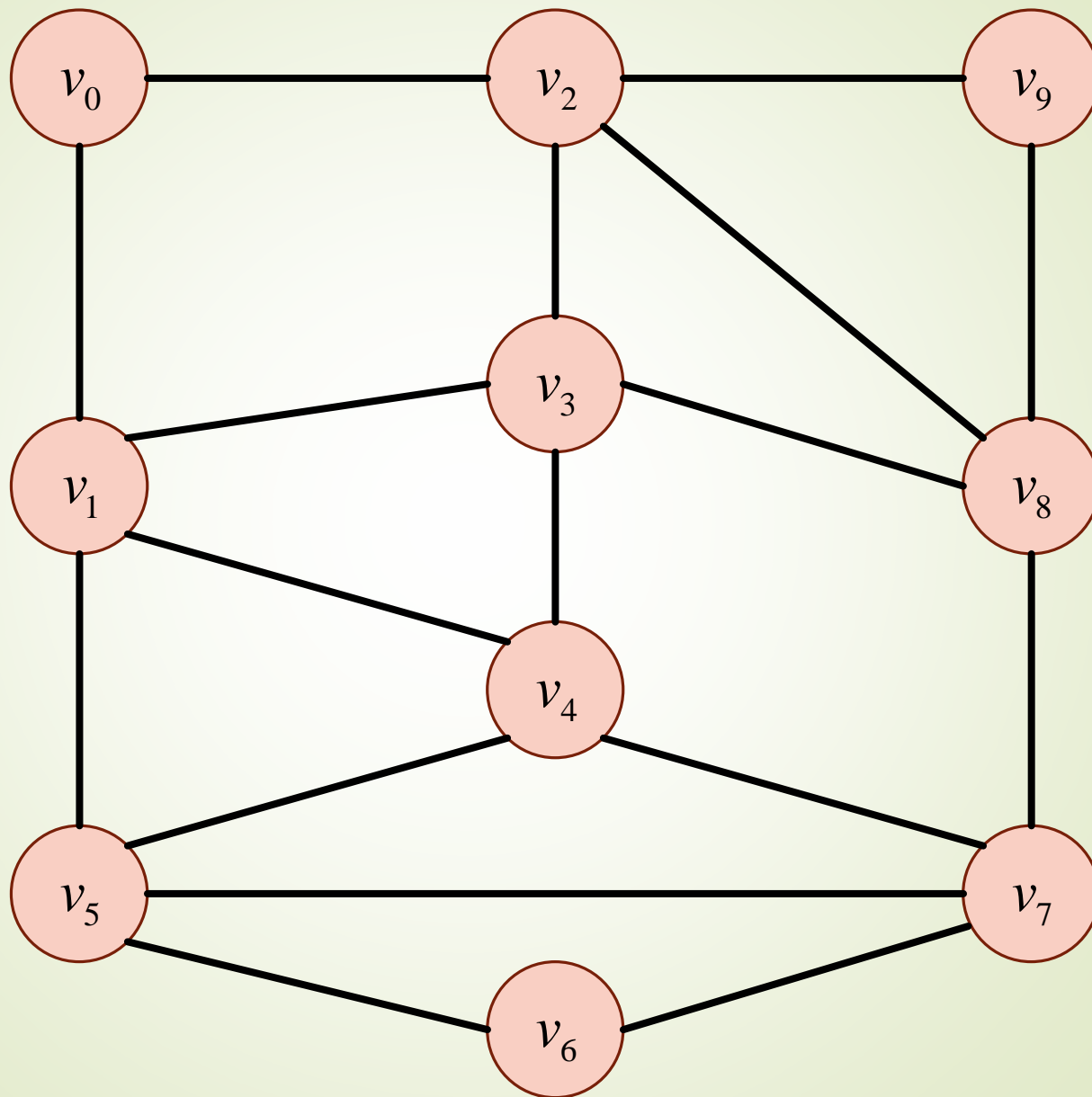
## 例

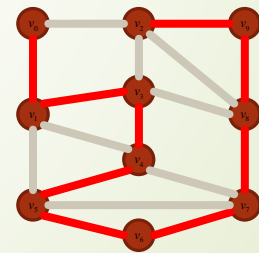
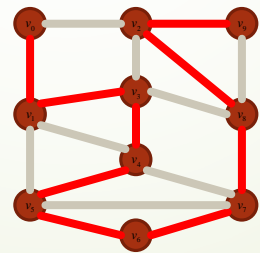
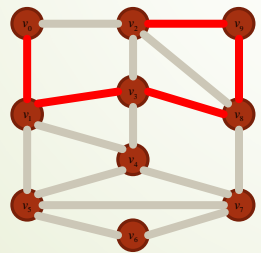
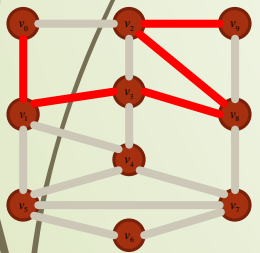
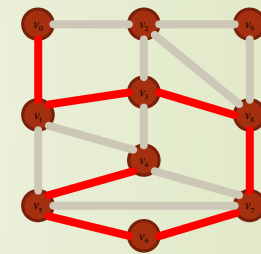
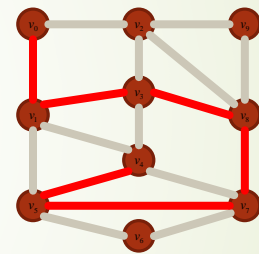
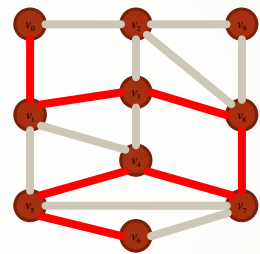
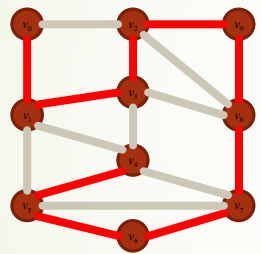
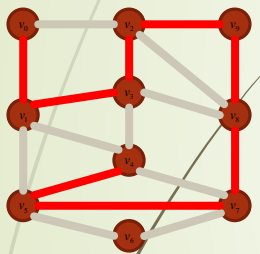
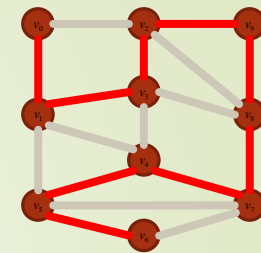
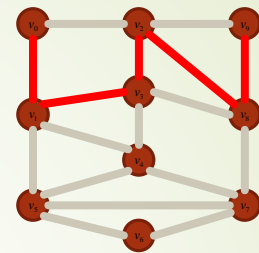
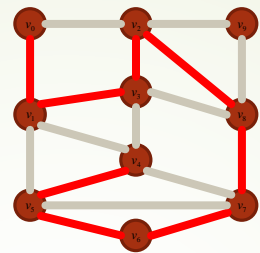
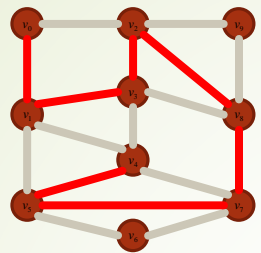
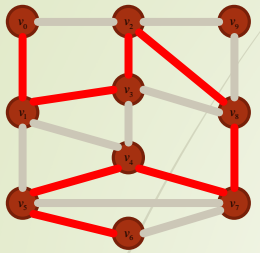


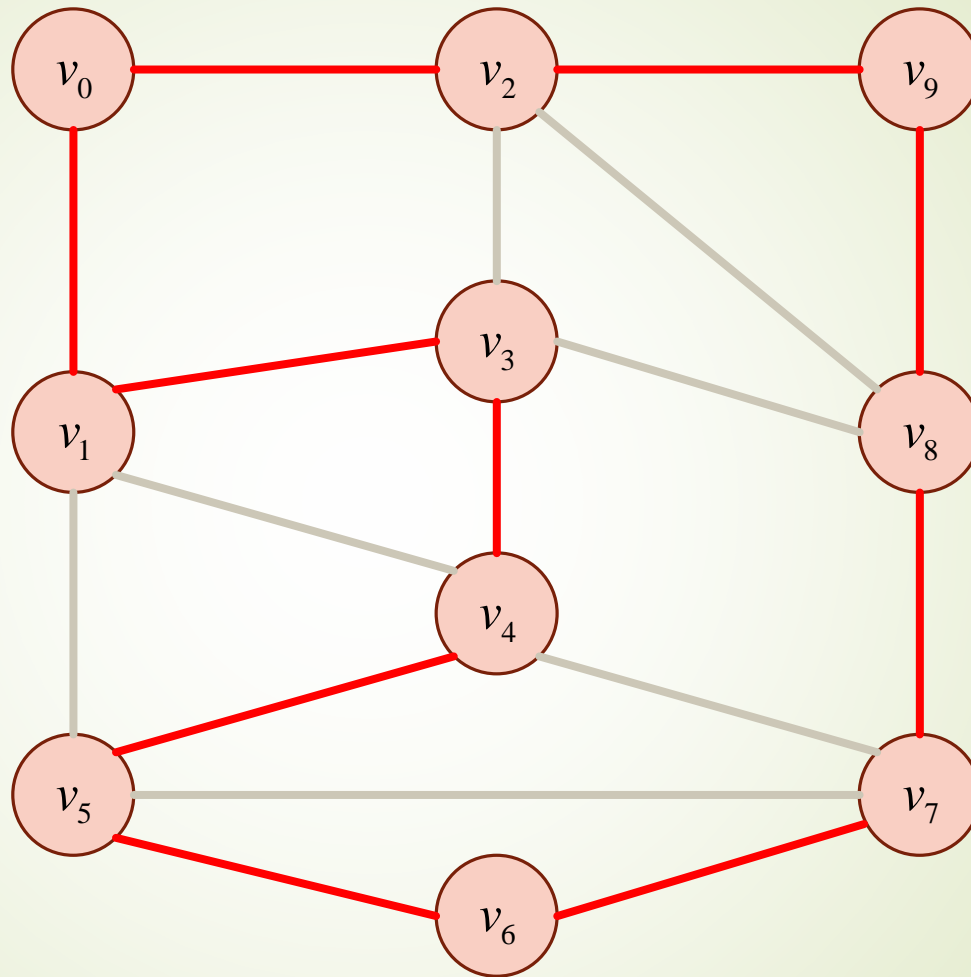


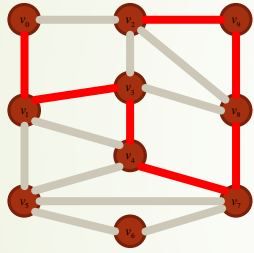
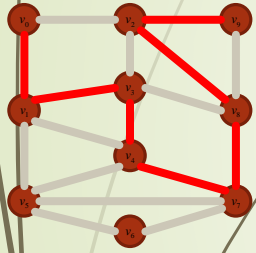
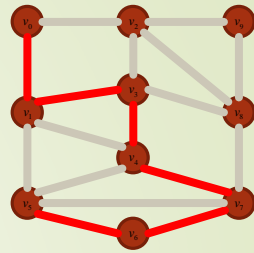
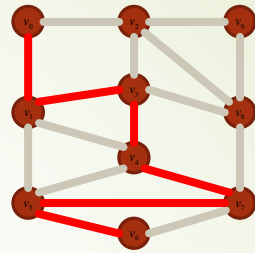
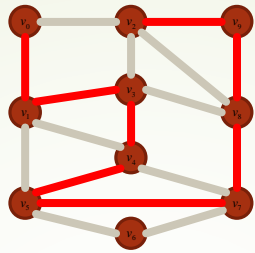
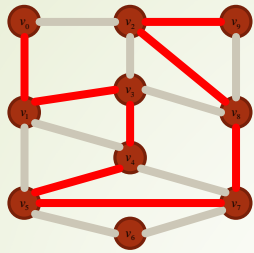
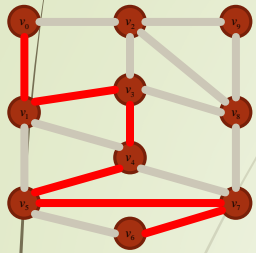


以下省略

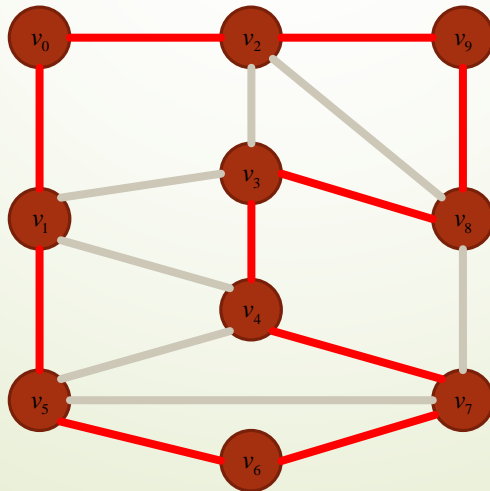








...



# 列挙のポイント

- 再帰アルゴリズム
  - 分岐点の位置に注意
- 分岐点に戻る際に、過去の探索履歴の消去
  - 深さ優先探索との違い

# 中間テスト

➡ 2018/6/7