

グラフの探索

離散数学・オートマトン

2024 年後期

佐賀大学工学部 只木進一

① 深さ優先探索 DFS: Depth-First search

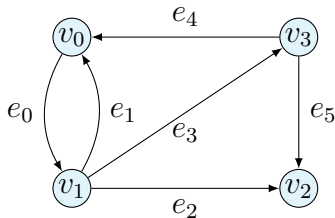
② DFA アルゴリズム

③ 幅優先探索 BFS: Breadth-First search

④ BFS アルゴリズム

有向グラフと探索

- 指定した頂点から、各頂点への経路を調べる

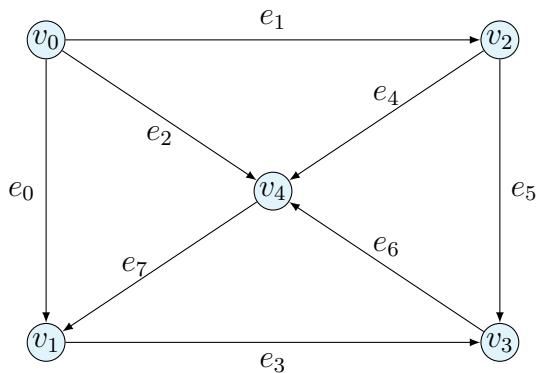


- 到達できるかに関心がある
- 適切なアルゴリズムを作る

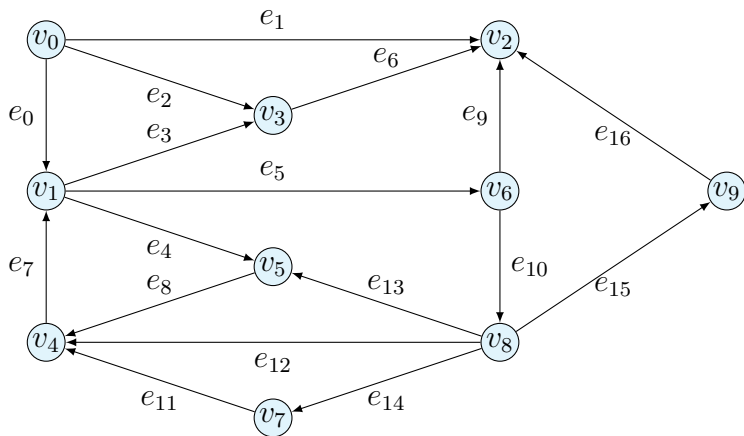
深さ優先探索 DFS: Depth-First search

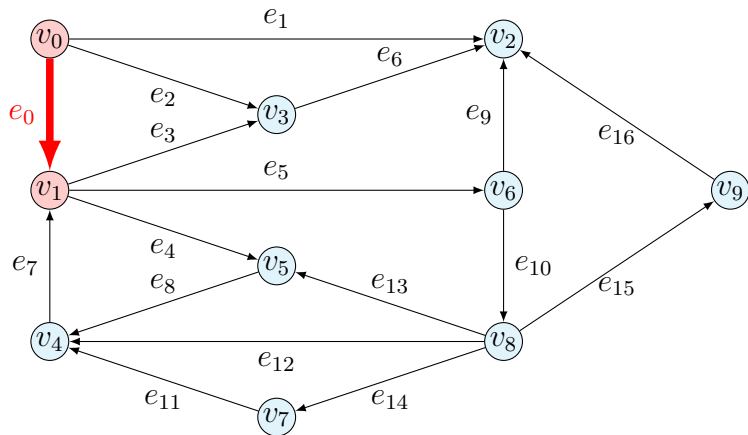
- 出発点を定める
- たどれる限り、辺をたどる: 再帰的アルゴリズム
 - それ以上進めなくなるまで
 - 新たな点が無くなるまで
- 道に戻って、分岐可能な頂点から、別の辺をたどる
- 結果としてできる木 (spanning trees) は、深いものができる

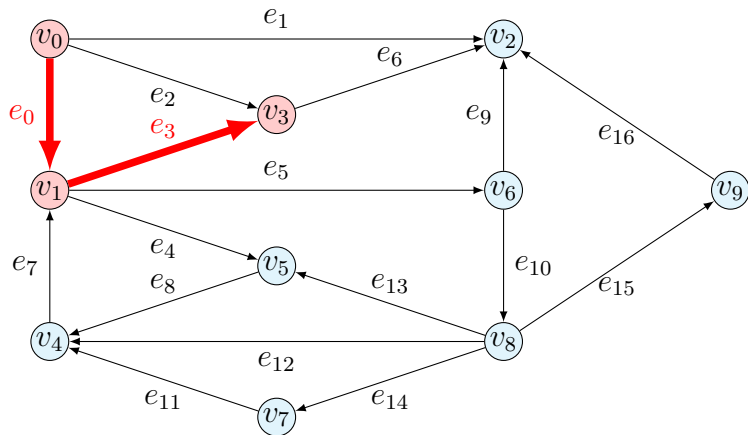
例 1.1:

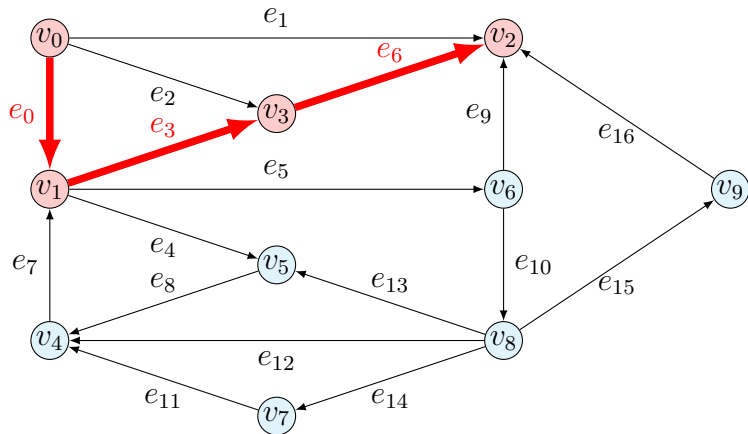


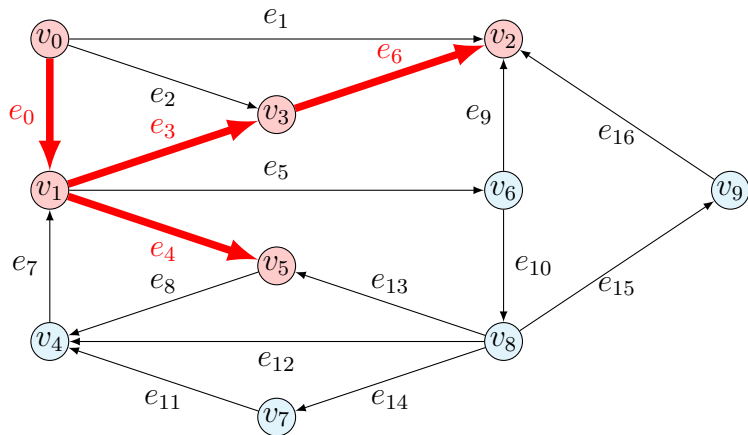
例 1.2:

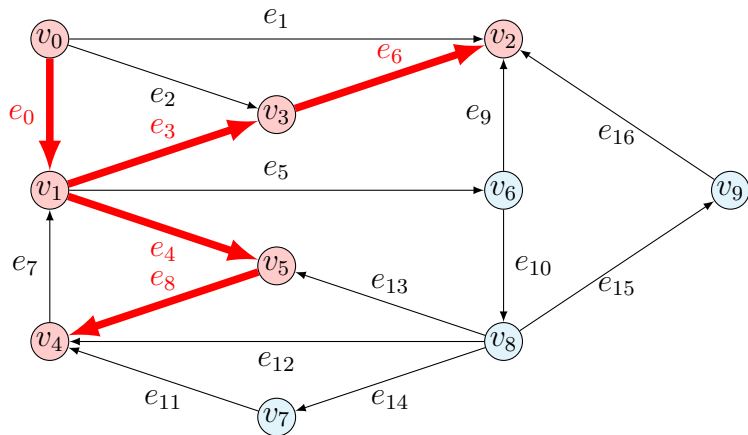


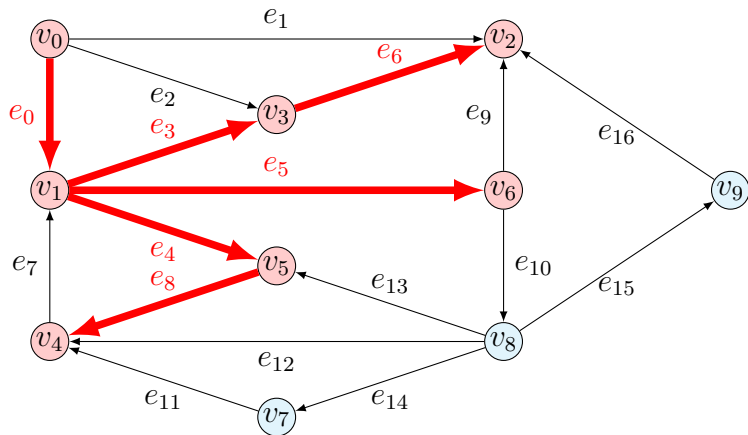


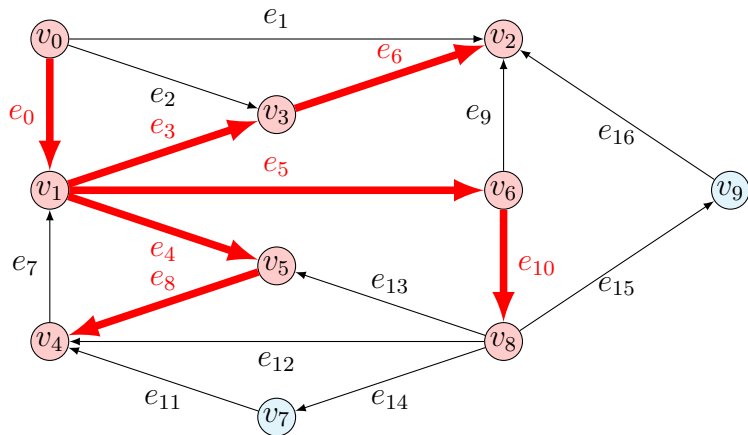


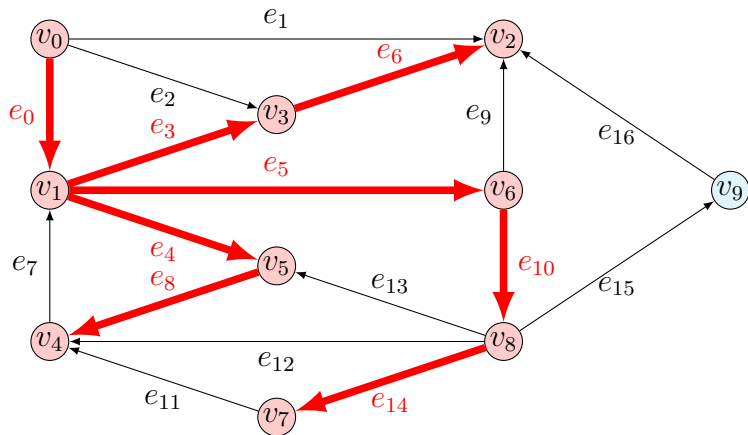


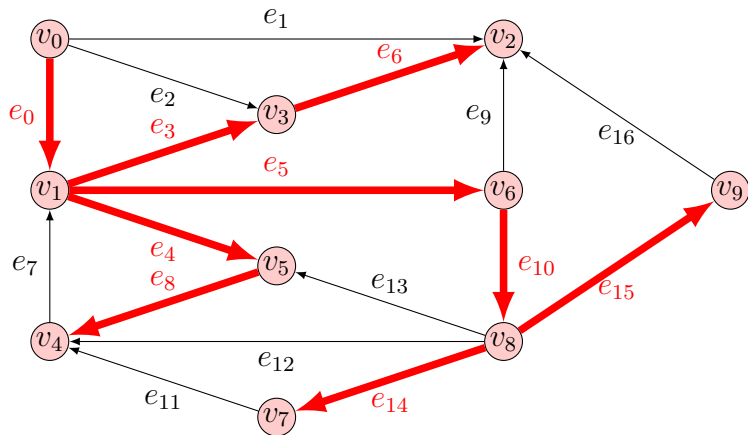












DFA アルゴリズム

- L : 到達した頂点のリスト: 初期値 $L = \{v_0\}$
- A : 使用する辺のリスト: 初期値 $A = \emptyset$

Algorithm 1 DFA アルゴリズム

procedure SEARCH(v, L, A)

for all $e \in \delta^+v$ **do**

▷ $//v$ から出る全ての辺

$w = \partial^-e$

if $w \notin L$ **then**

$L.append(w)$

$A.append(e)$

 SEARCH(w, L, A)

end if

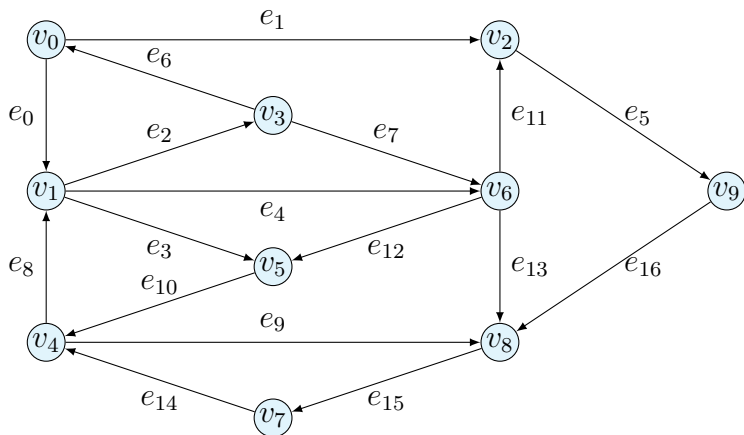
end for

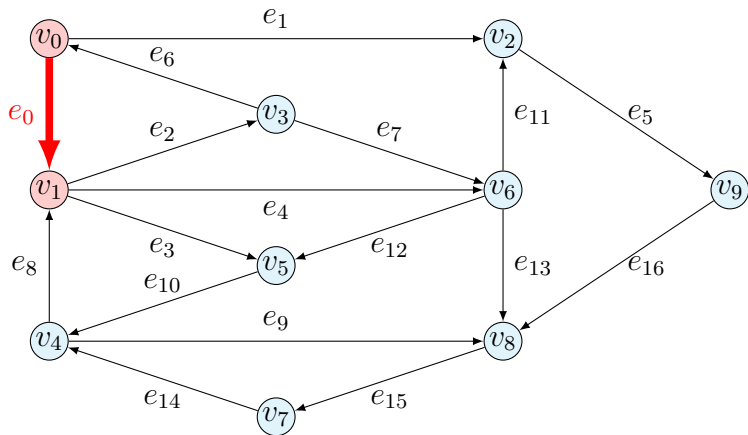
end procedure

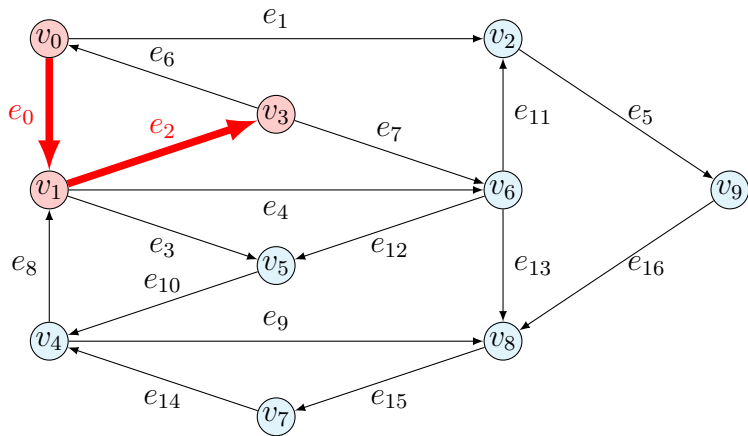
実行状況

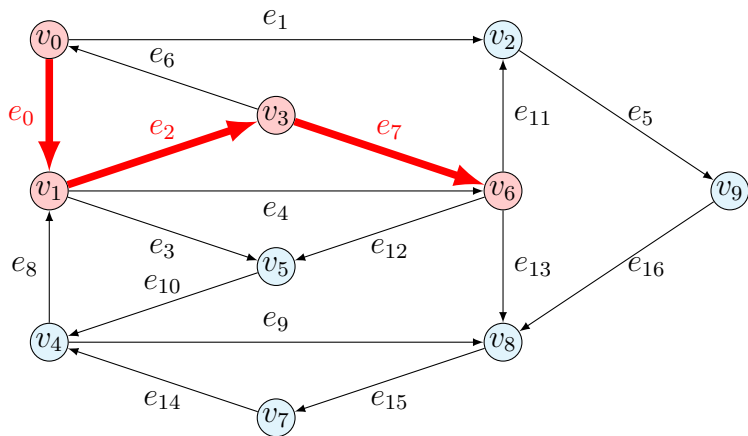
$(v_0, [v_0], []) \rightarrow (v_1, [v_0, v_1], [e_0])$
 $\rightarrow (v_3, [v_0, v_1, v_3], [e_0, e_3])$
 $\rightarrow (v_2, [v_0, v_1, v_3, v_2], [e_0, e_3, e_6])$
 $\rightarrow (v_5, [v_0, v_1, v_3, v_2, v_5], [e_0, e_3, e_6, e_4])$
 $\rightarrow (v_4, [v_0, v_1, v_3, v_2, v_5, v_4], [e_0, e_3, e_6, e_4, e_8])$
 $\rightarrow (v_6, [v_0, v_1, v_3, v_2, v_5, v_4, v_6], [e_0, e_3, e_6, e_4, e_8, e_5])$
 $\rightarrow (v_8, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8], [e_0, e_3, e_6, e_4, e_8, e_5, e_{10}])$
 $\rightarrow (v_7, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8, v_7], [e_0, e_3, e_6, e_4, e_8, e_5, e_{10}, e_{14}])$
 $\rightarrow (v_9, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8, v_7, v_9], [e_0, e_3, e_6, e_4, e_8, e_5, e_{10}, e_{14}, e_{15}])$

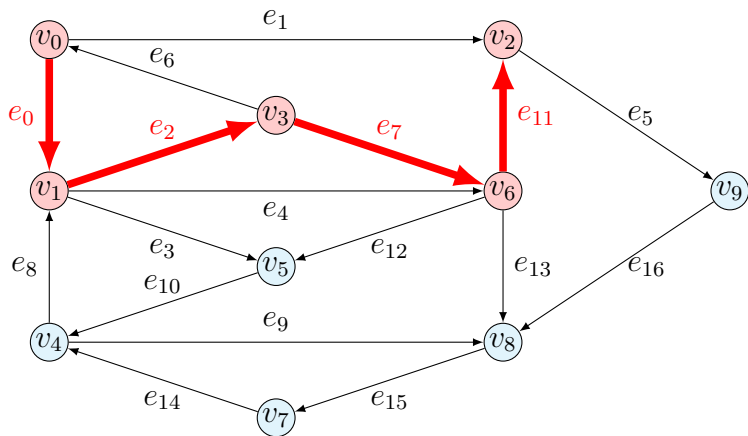
例 2.1:

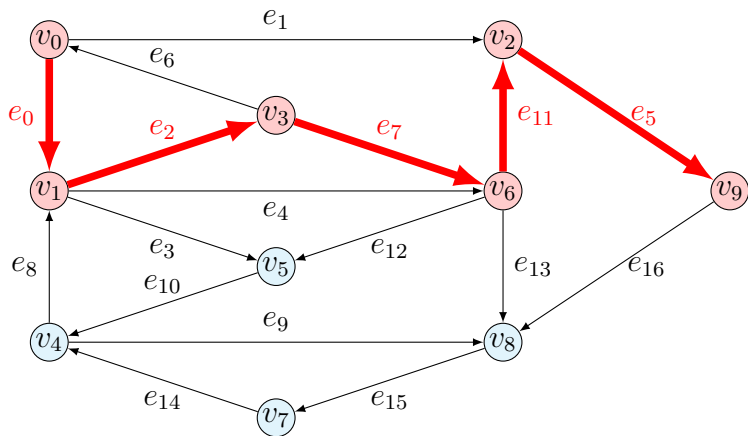


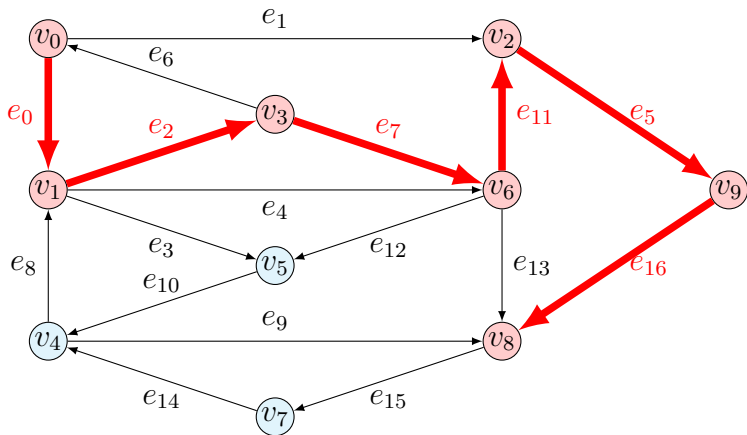


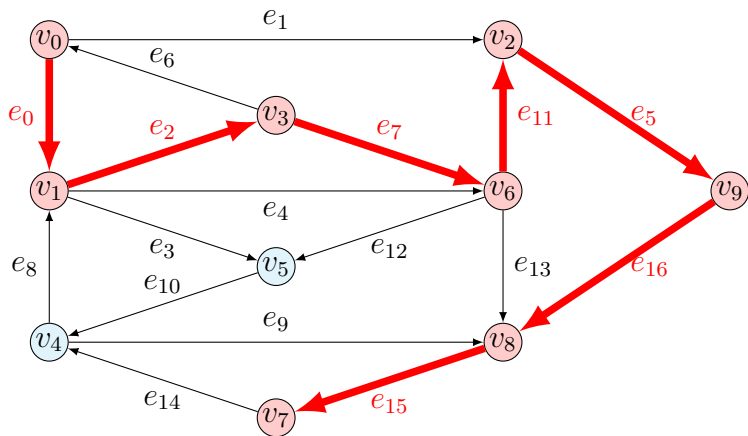


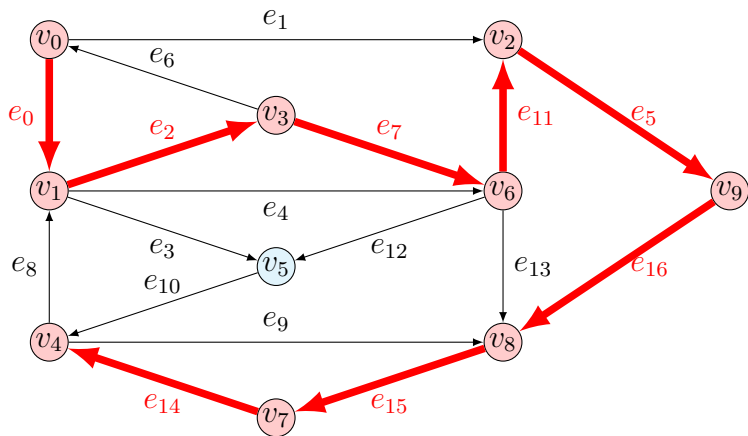


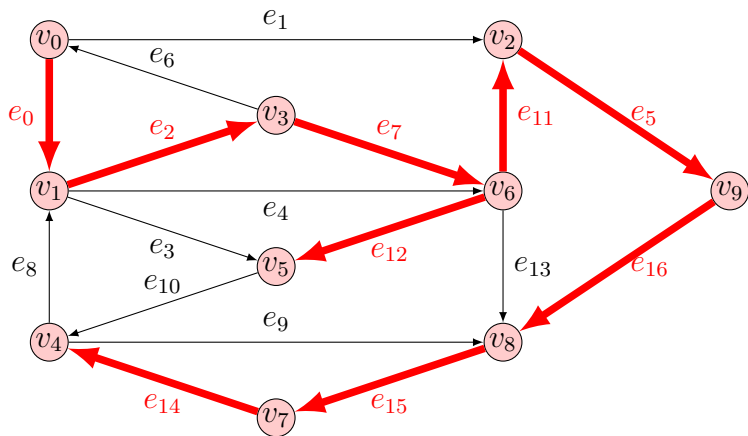








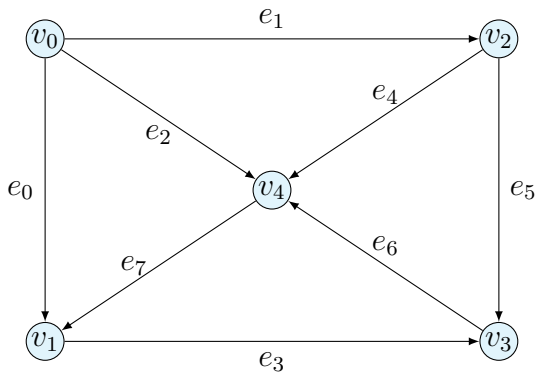




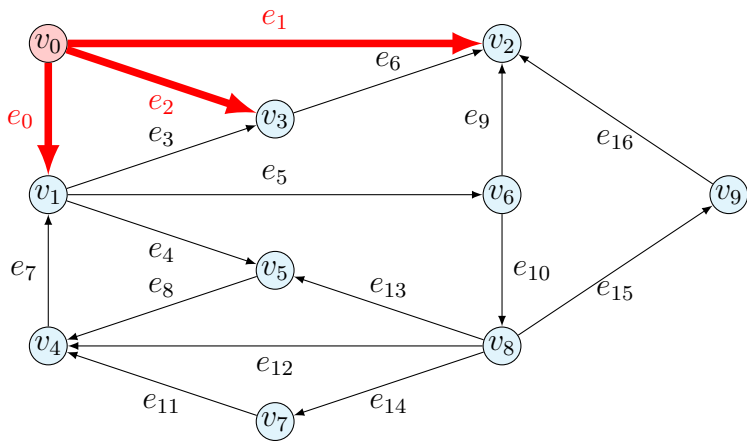
幅優先探索 BFS: Breadth-First search

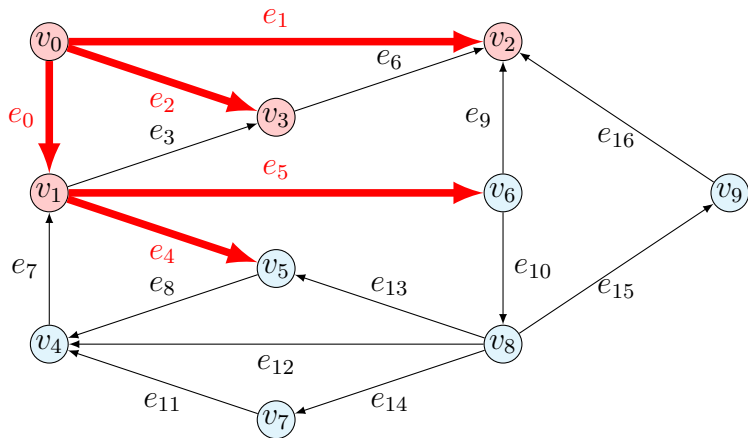
- 出発点を定める。この点の集合を S_0 とする。
- 新たな頂点がなくなるまで繰り返す
 - S_{i-1} の各点の隣接頂点のうち、未調査の点の集合を S_i とする
- 出発点から、移動する回数を一つずつ増やしていく
- 一つの頂点への複数の経路がある場合に注意
- 結果としてできる木 (spanning tree) は、浅いものができる

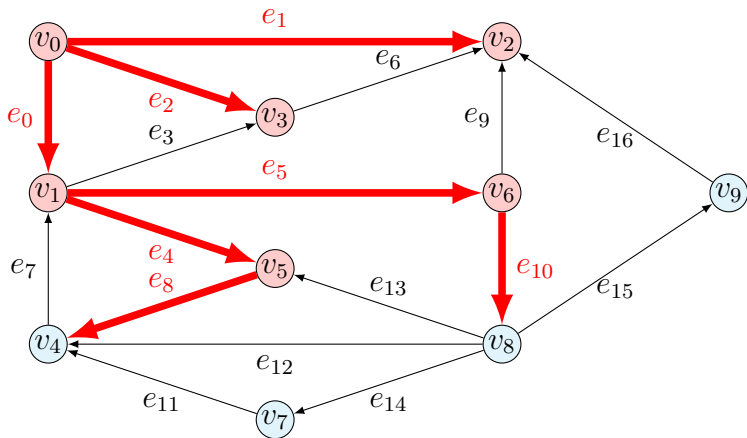
例 3.1:

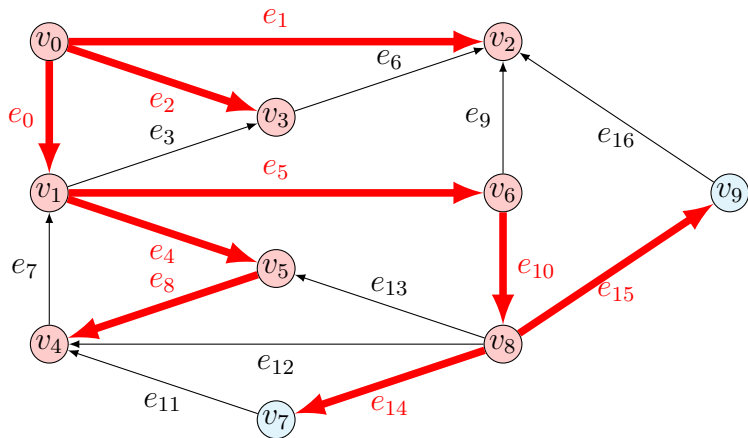


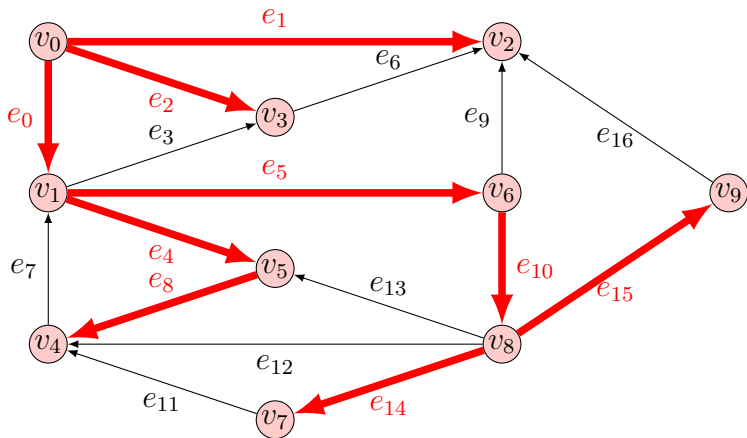
例 1.2 に対する BFS の結果











BFS アルゴリズム

Algorithm 2 BFS アルゴリズム

```
 $L = \emptyset$   
 $A = \emptyset$   
 $Q = [(\text{Null}, r)]$   
while  $Q \neq \emptyset$  do  
     $(s, v) = e = Q.\text{poke}()$   
    if  $v \notin L$  then  
        if  $s \neq \text{Null}$  then  
             $A.\text{append}(e)$   
        end if  
        for all  $e \in \delta^+ v$  do  
            if  $e \notin Q$  then  
                 $Q.\text{push}(e)$   
            end if  
        end for  
         $L.\text{append}(v)$   
    end if  
end while
```

- ▷ 到達済み頂点のリスト
- ▷ 探索木の辺のリスト
- ▷ 調査すべき辺の待ち行列

- ▷ 待ち行列の先頭要素を取り出す

待ち行列: Queue

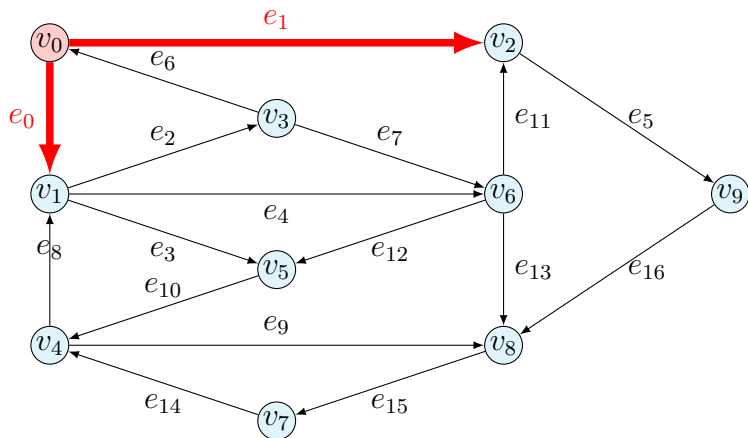
- リストの一種
- 末尾から要素を追加
- 先頭から要素を削除
- First-In-First-Out

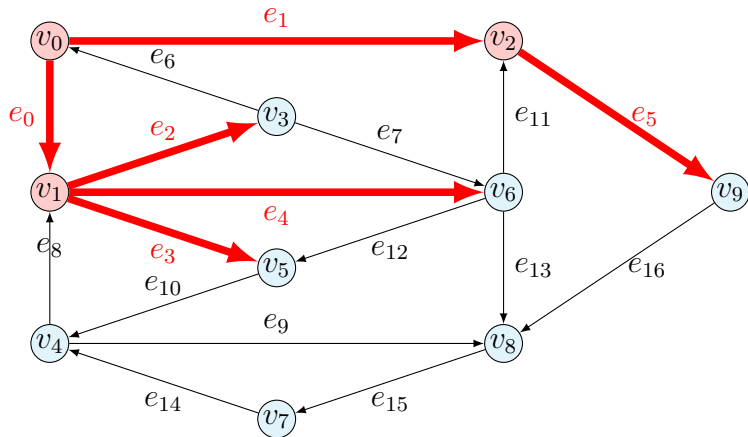


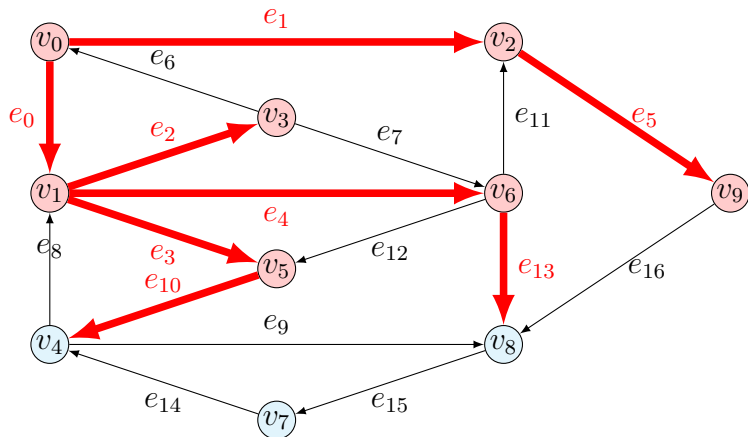
例 1.2: 探索の状況

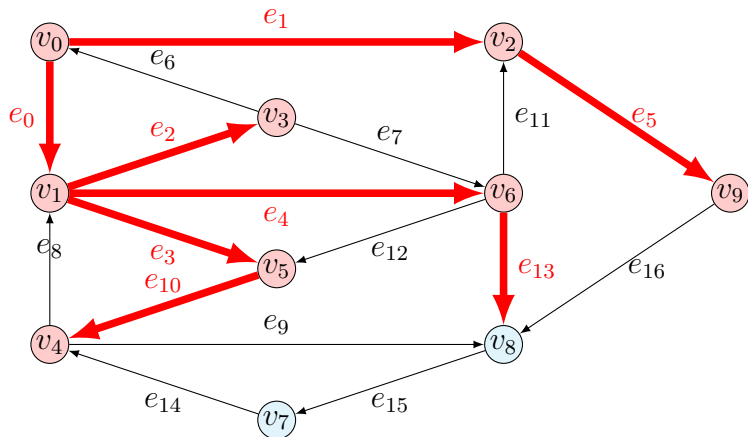
	現在の頂点	L	Q
0		\emptyset	$[, v_0]$
1	v_0	$[v_0]$	$[e_0, e_1, e_2]$
2	v_1	$[v_0, v_1]$	$[e_1, e_2, e_4, e_5]$
3	v_2	$[v_0, v_1, v_2]$	$[e_2, e_4, e_5]$
4	v_3	$[v_0, v_1, v_2, v_3]$	$[e_4, e_5]$
5	v_5	$[v_0, v_1, v_2, v_3, v_5]$	$[e_5, e_8]$
6	v_6	$[v_0, v_1, v_2, v_3, v_5, v_6]$	$[e_8, e_{10}]$
7	v_4	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4]$	$[e_{10}]$
8	v_8	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8]$	$[e_{14}, e_{15}]$
9	v_7	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8, v_7]$	$[e_{15}]$
10	v_9	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8, v_7, v_9]$	$[]$

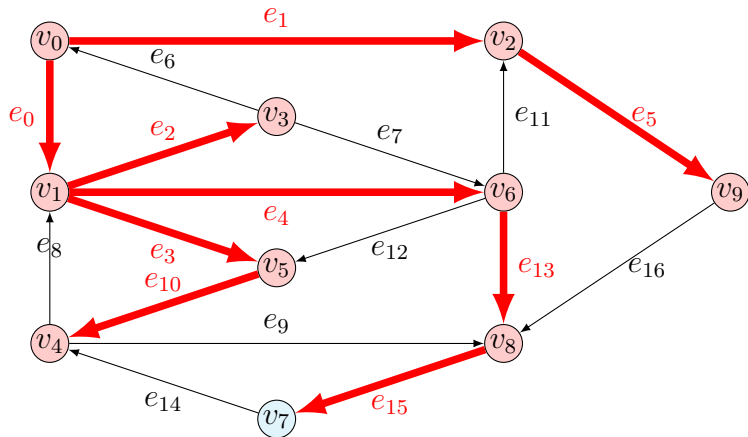
例 2.1: 結果

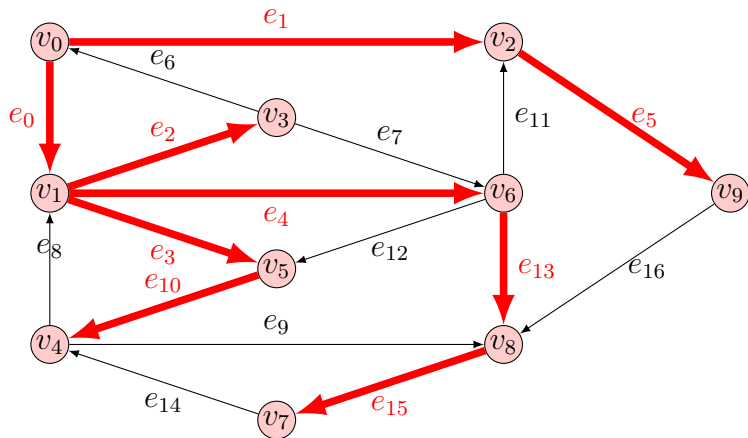












例 2.1: 探索の状況

	現在の頂点	L	Q
0		\emptyset	$[, v_0]$
1	v_0	$[v_0]$	$[e_0, e_1]$
2	v_1	$[v_0, v_1]$	$[e_1, e_2, e_3, e_4]$
3	v_2	$[v_0, v_1, v_2]$	$[e_2, e_3, e_4, e_5]$
4	v_3	$[v_0, v_1, v_2, v_3]$	$[e_3, e_4, e_5]$
5	v_5	$[v_0, v_1, v_2, v_3, v_5]$	$[e_4, e_5, e_{10}]$
6	v_6	$[v_0, v_1, v_2, v_3, v_5, v_6]$	$[e_5, e_{10}, e_{13}]$
7	v_9	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9]$	$[e_{10}, e_{13}]$
8	v_4	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4]$	$[e_{13}]$
9	v_8	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4, v_8]$	$[e_{15}]$
10	v_7	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4, v_8, v_7]$	$[]$