

# グラフの探索

離散数学・オートマトン

2022 年後期

佐賀大学理工学部 只木進一

① 深さ優先探索 DFS: Depth-First Search

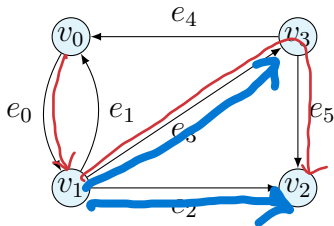
再帰

② 幅優先探索 BFS: Breadth-First Search

待ち行列

# 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

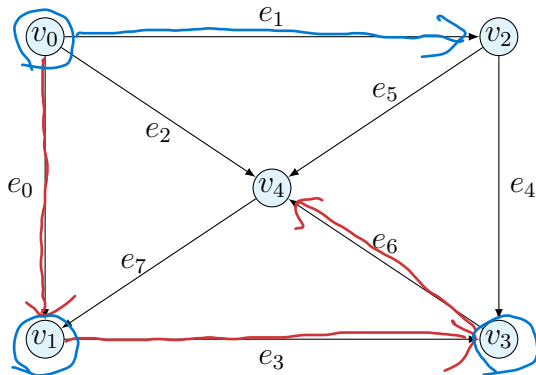


適切なアルゴリズムを作る

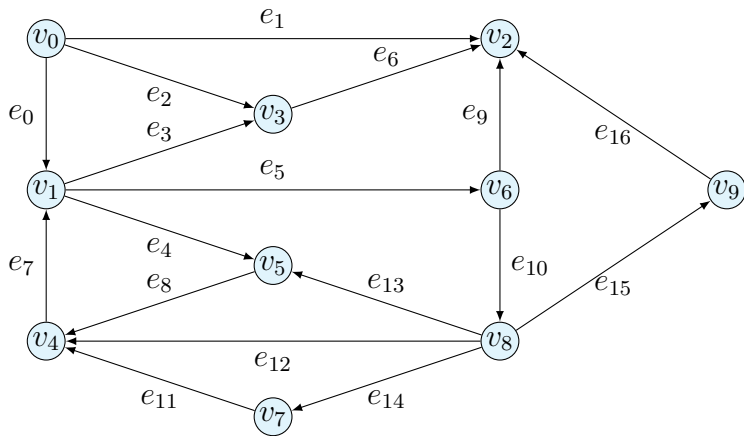
# 深さ優先探索 DFS: Depth-First Search

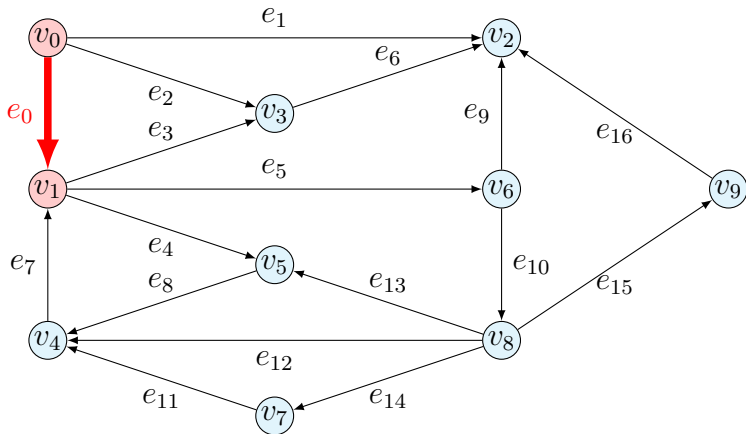
- ✓ 出発点を定める
- ✓ たどれる限り、辺をたどる: 再帰的アルゴリズム
  - それ以上進めなくなるまで
  - 新たな点が無くなるまで
- 道に戻って、別の辺をたどる
- 結果としてできる木 (spanning tree) は、深いものができる  
始点から遠い

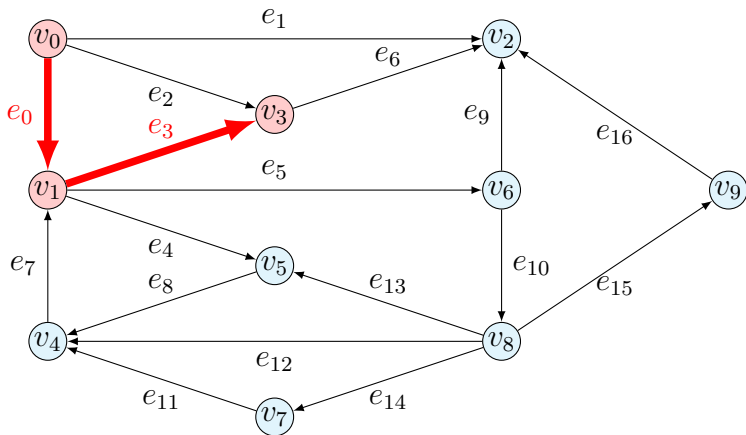
## 例 1.1:



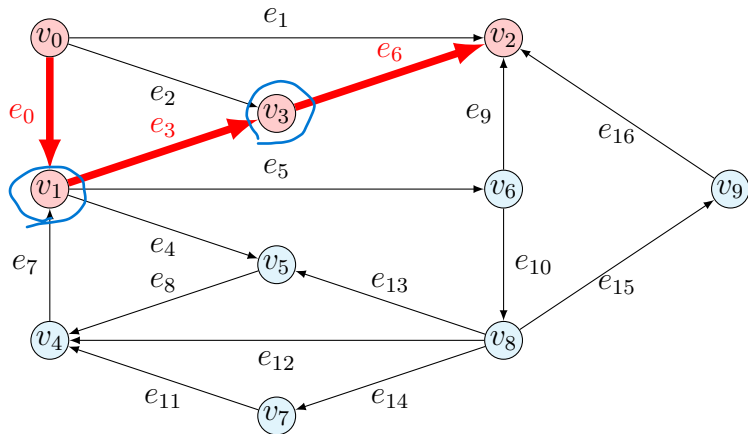
## 例 1.2:

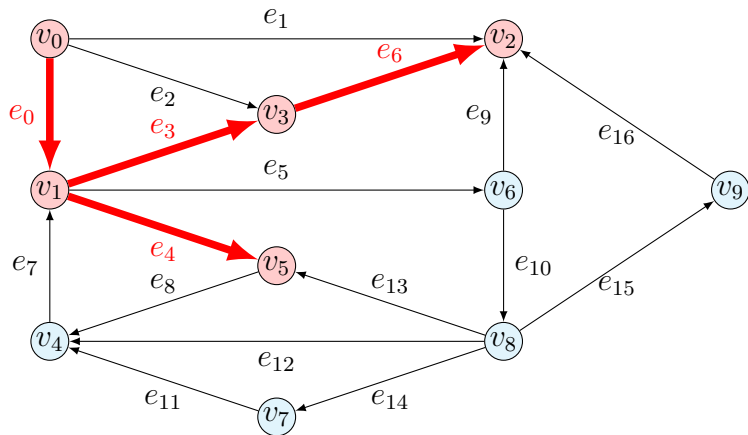


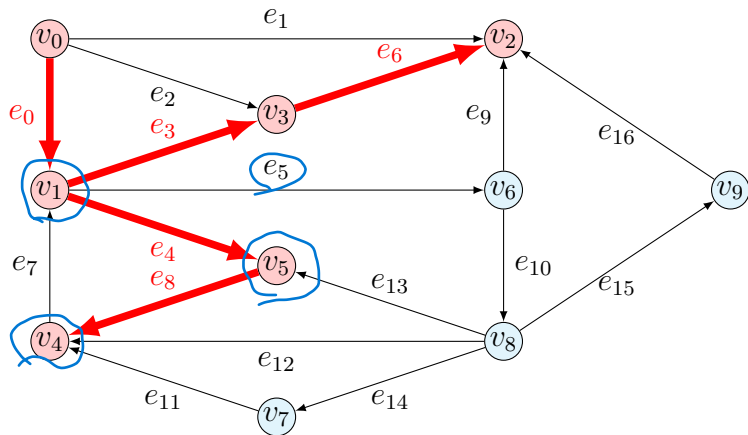


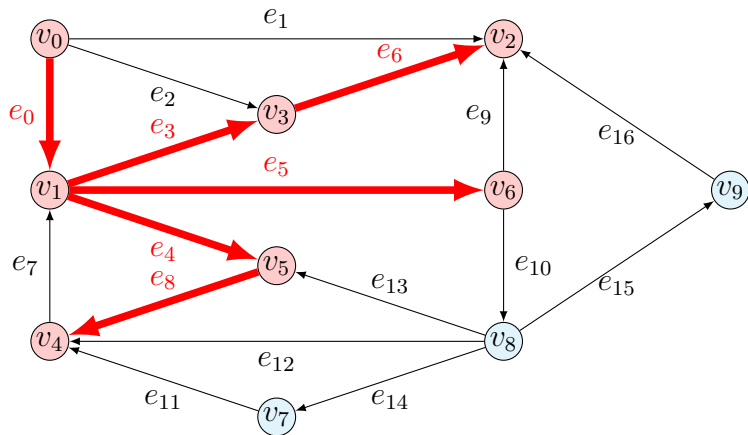


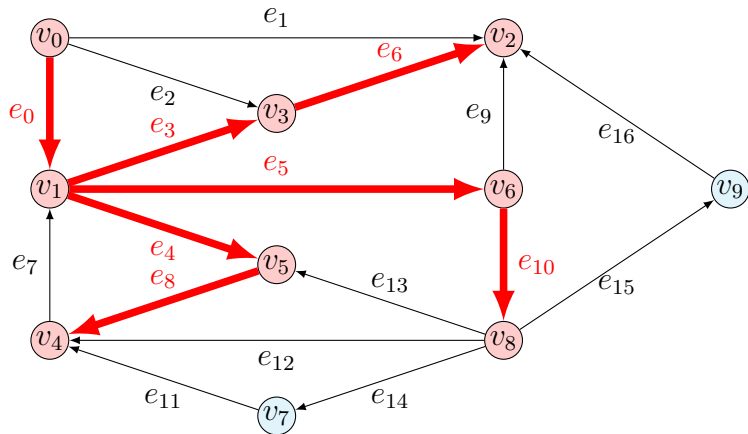


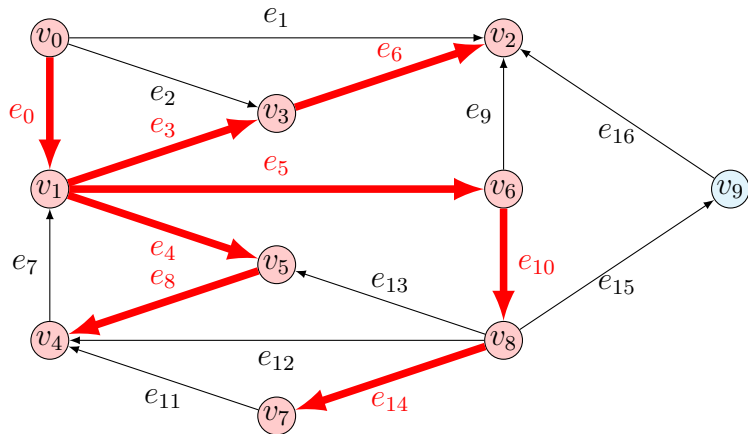


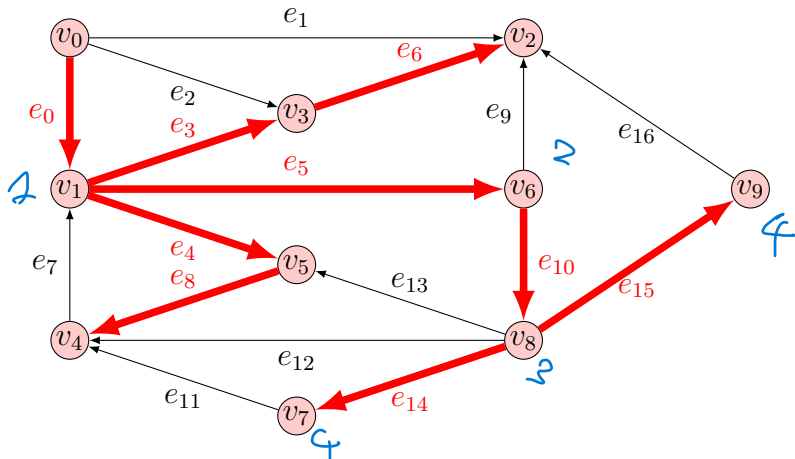












---

**Algorithm 1 DFA アルゴリズム**


---

**procedure** SEARCH( $v, L$ )

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

$w = \partial^-e$  : 反対側

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

$L.append(w)$

SEARCH( $w, L$ )

**end if**

**end for**

**end procedure**

---

$v$ : 注目している頂点、

$\triangleright //v$  から出る全ての辺

これまで訪ねた頂点のリスト  
(集合)

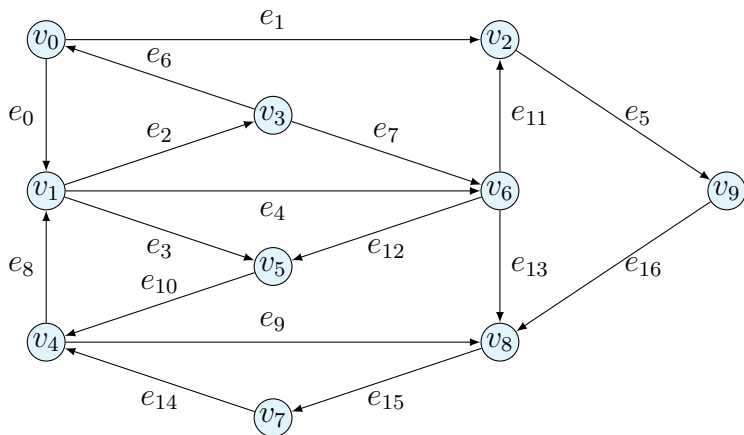
、分枝を順に試す

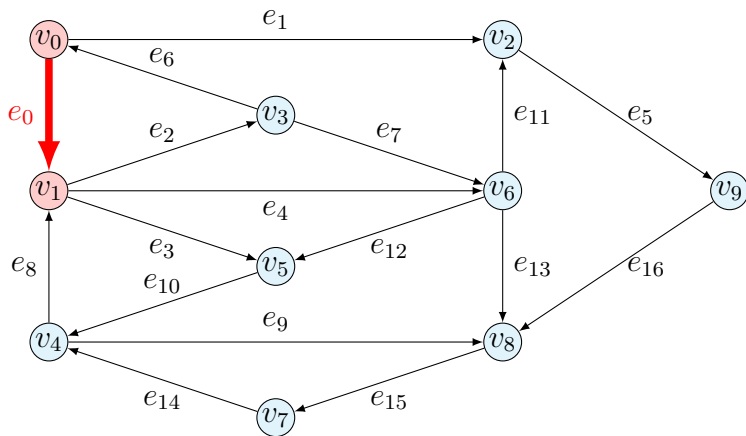


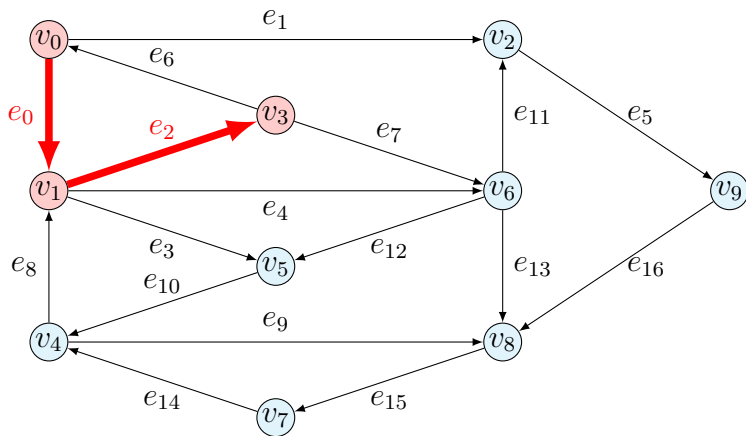
# 実行状況

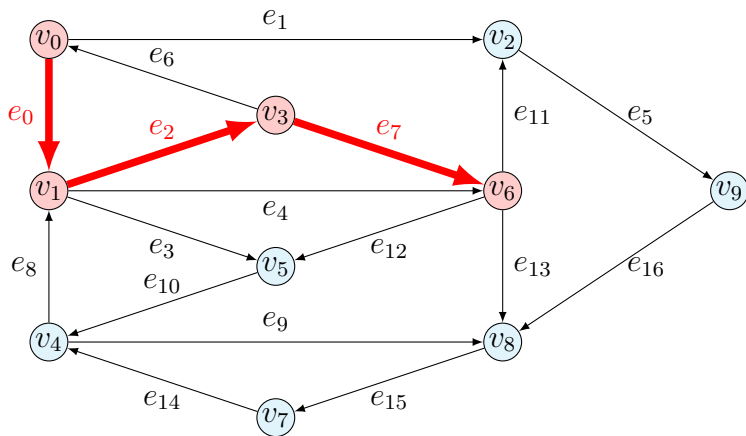
$$\begin{aligned}(v_0, [v_0]) &\rightarrow (v_1, [v_0, v_1]) \\ &\rightarrow (v_3, [v_0, v_1, v_3]) \\ &\rightarrow (v_2, [v_0, v_1, v_3, v_2]) \\ &\rightarrow (v_5, [v_0, v_1, v_3, v_2, v_5]) \\ &\rightarrow (v_4, [v_0, v_1, v_3, v_2, v_5, v_4]) \\ &\rightarrow (v_6, [v_0, v_1, v_3, v_2, v_5, v_4, v_6]) \\ &\rightarrow (v_8, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8]) \\ &\rightarrow (v_7, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8, v_7]) \\ &\rightarrow (v_9, [v_0, v_1, v_3, v_2, v_5, v_4, v_6, v_8, v_7, v_9])\end{aligned}$$

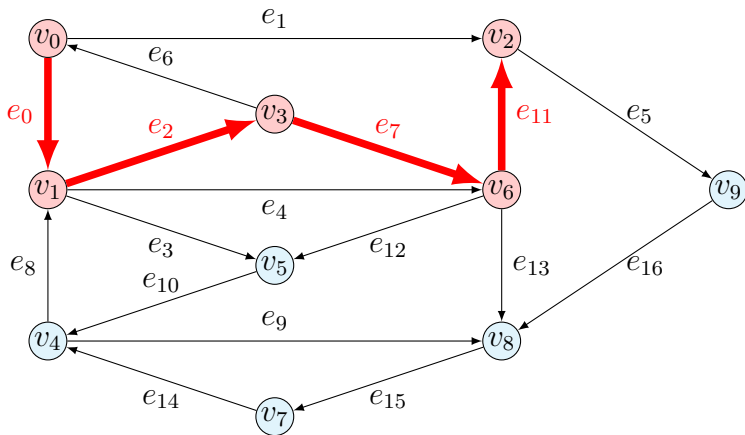
## 例 1.3:

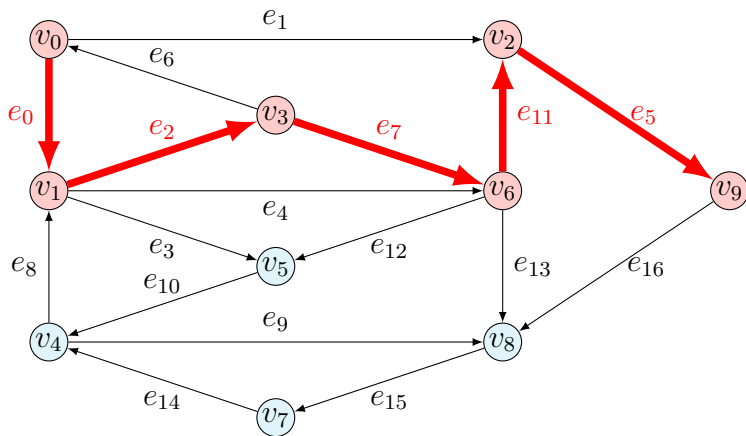


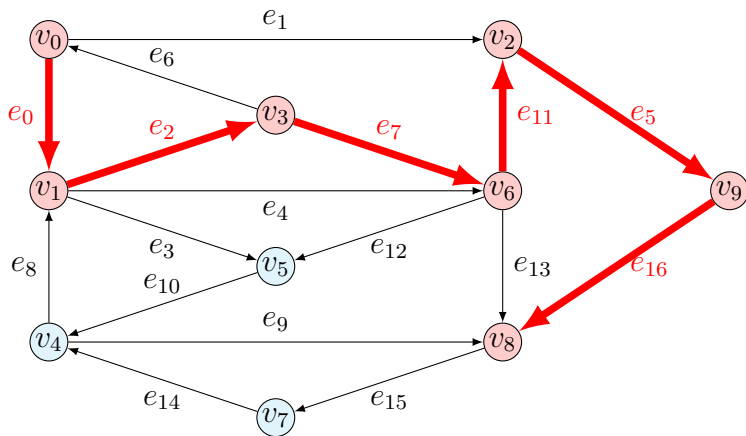




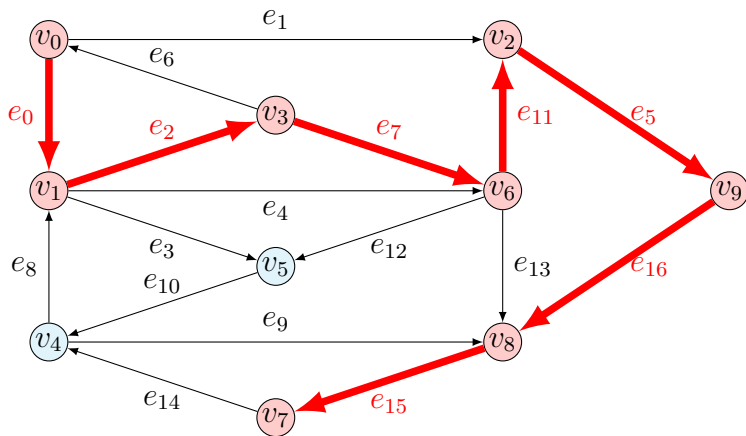


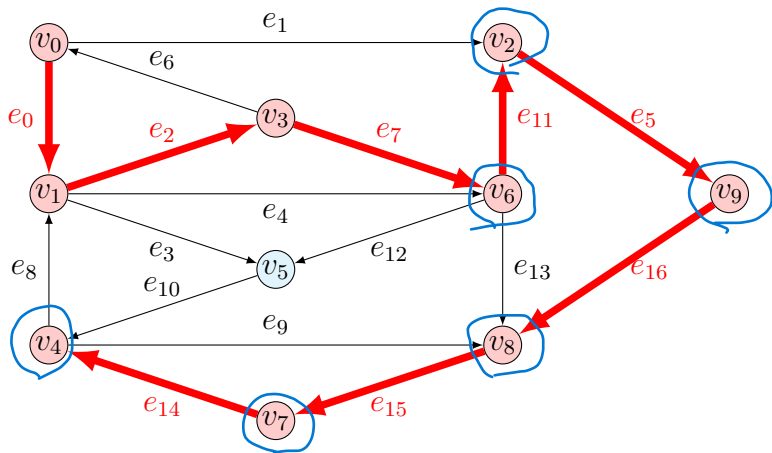


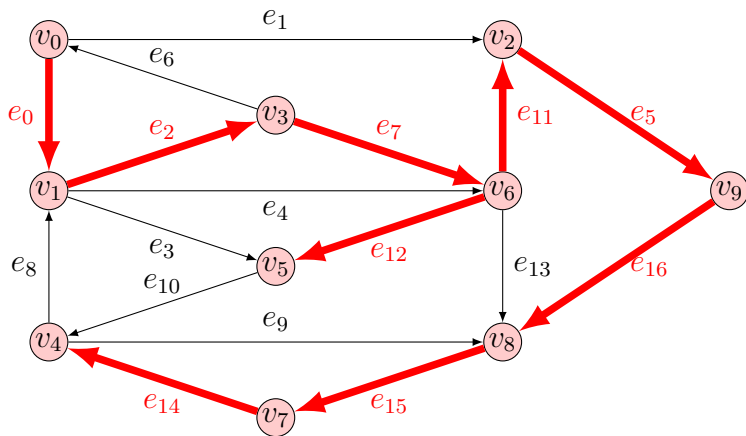










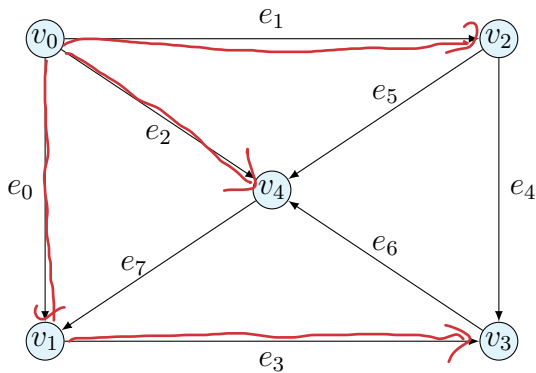


# 幅優先探索 BFS: Breadth-First Search

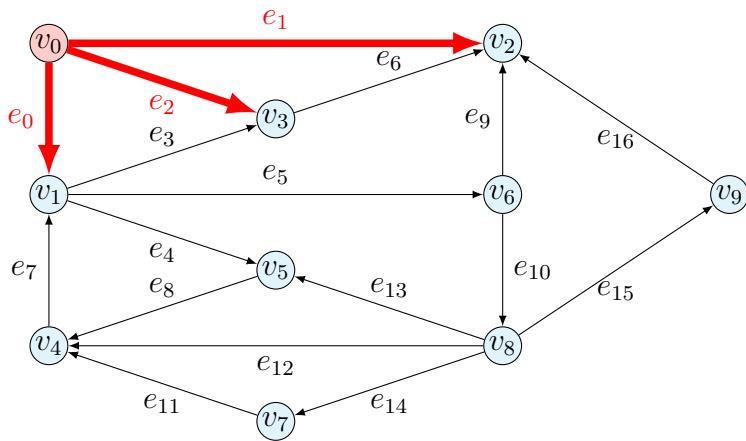
Breath

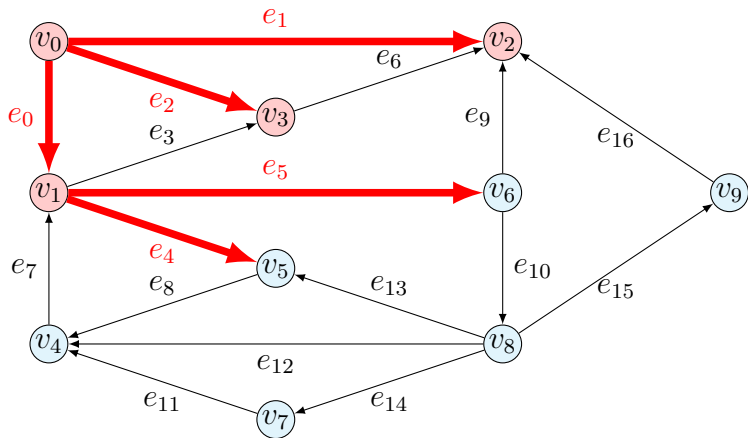
- 出発点を定める。この点の集合を  $S_0$  とする。
- 新たな頂点がなくなるまで繰り返す
  - $S_{i-1}$  の各点の隣接頂点のうち、未調査の点の集合を  $S_i$  とする
- 結果としてできる木 (spanning tree) は、浅いものができる

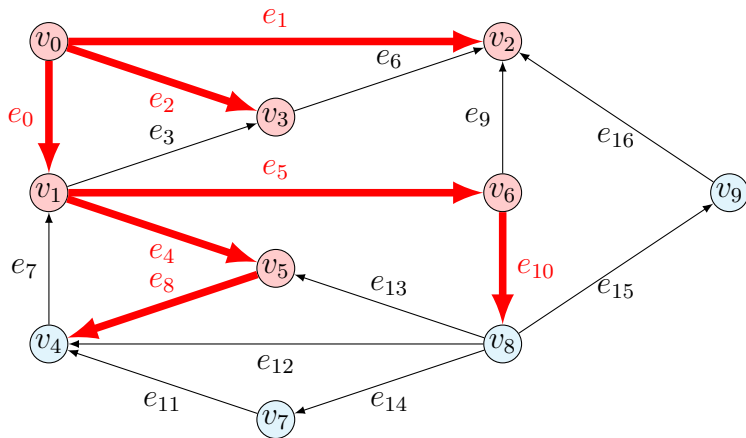
## 例 2.1:



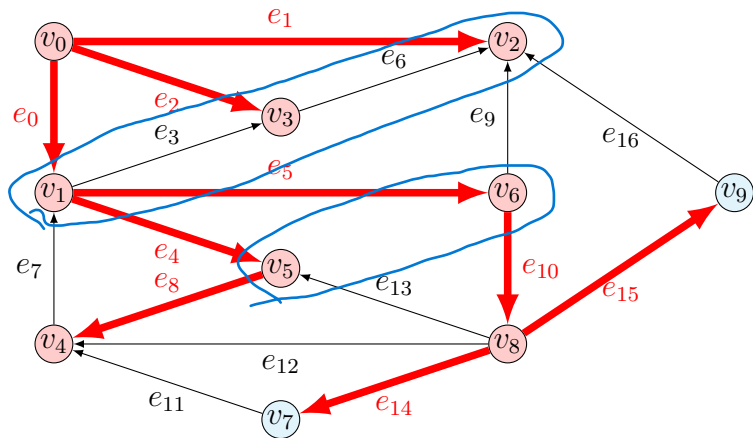
## 例 1.2 に対する BFS の結果

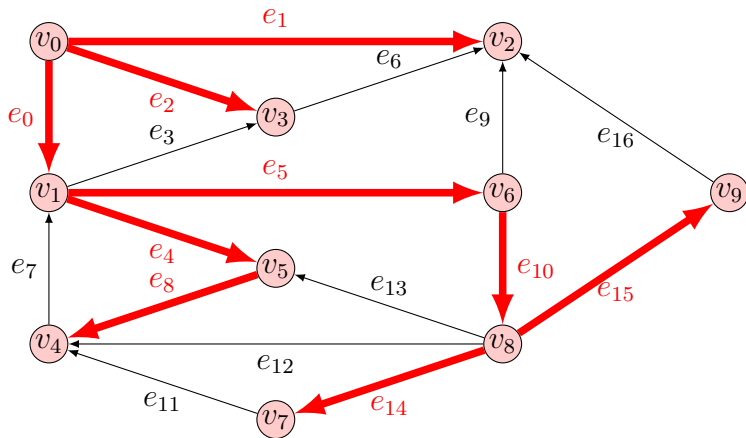












---

## Algorithm 2 BFS アルゴリズム

---

 $L = \emptyset$ 
 $Q = [r]$ 
**while**  $Q \neq \emptyset$  **do**
 $v = Q.\text{poke}()$  ←

**for all**  $e \in \delta^+v$  **do**
 $w = \partial^-e$  ←

**if**  $w \notin L \wedge w \notin Q$  **then**
 $Q.\text{push}(w)$ 
**end if**
**end for**
 $L.\text{append}(w)$  ←

**end while**


---

▷ 到達済み頂点のリスト

▷ 調査すべき頂点の待ち行列

▷ 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

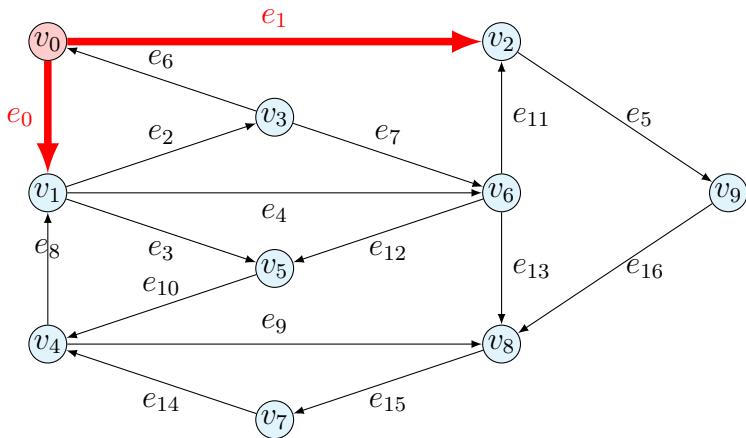
cf: stack

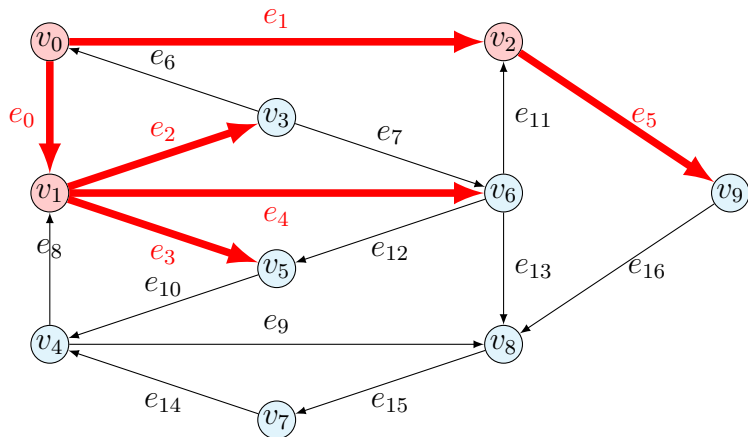


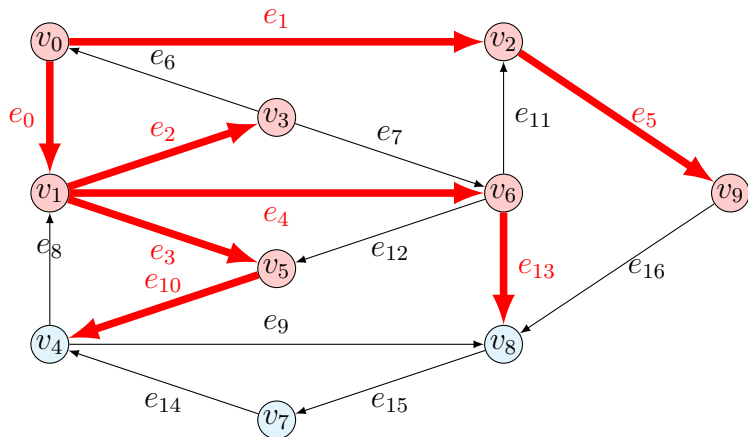
## 例 1.2: 探索の状況

	現在の頂点	$L$	$Q$
0		$\emptyset$	$[v_0]$
1	$v_0$	$[v_0]$	$[v_1, v_2, v_3]$
2	$v_1$	$[v_0, v_1]$	$[v_2, v_3, v_5, v_6]$
3	$v_2$	$[v_0, v_1, v_2]$	$[v_3, v_5, v_6]$
4	$v_3$	$[v_0, v_1, v_2, v_3]$	$[v_5, v_6]$
5	$v_5$	$[v_0, v_1, v_2, v_3, v_5]$	$[v_6, v_4]$
6	$v_6$	$[v_0, v_1, v_2, v_3, v_5, v_6]$	$[v_4, v_8]$
7	$v_4$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4]$	$[v_8]$
8	$v_8$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8]$	$[v_7, v_9]$
9	$v_7$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8, v_7]$	$[v_9]$
10	$v_9$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_4, v_8, v_7, v_9]$	$\emptyset$

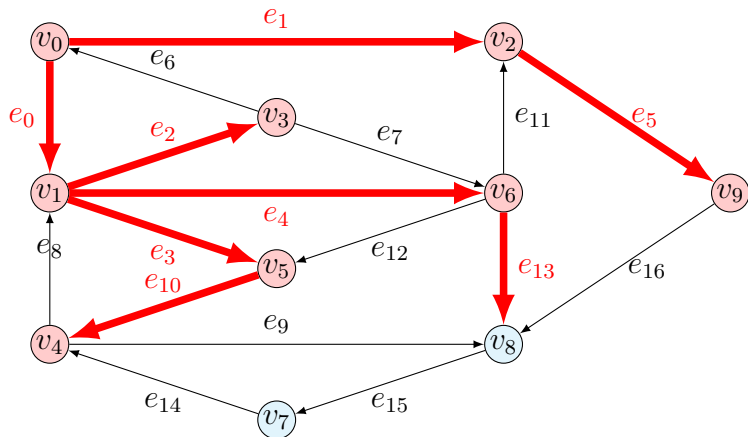
## 例 1.3: 結果

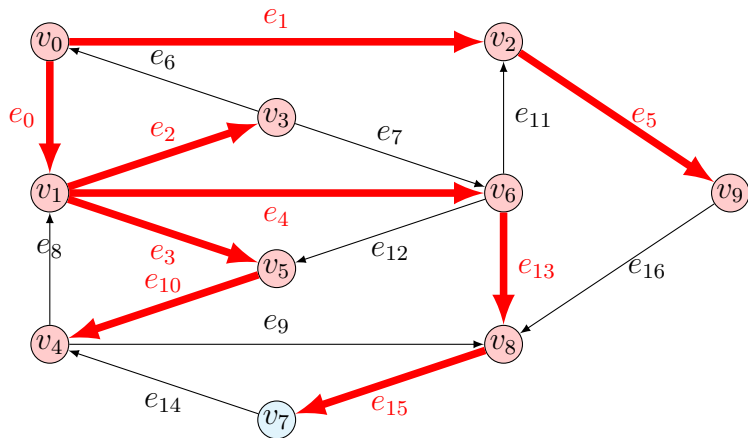


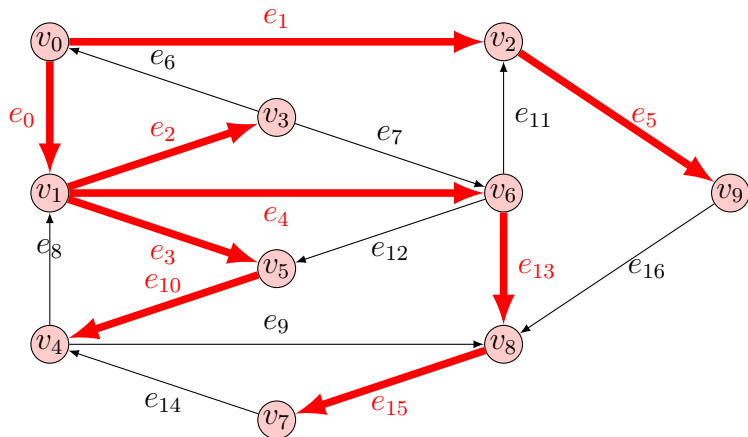












## 例 1.3: 探索の状況

	現在の頂点	$L$	$Q$
0		$\emptyset$	$[v_0]$
1	$v_0$	$[v_0]$	$[v_1, v_2]$
2	$v_1$	$[v_0, v_1]$	$[v_2, v_3, v_5, v_6]$
3	$v_2$	$[v_0, v_1, v_2]$	$[v_3, v_5, v_6, v_9]$
4	$v_3$	$[v_0, v_1, v_2, v_3]$	$[v_5, v_6, v_9]$
5	$v_5$	$[v_0, v_1, v_2, v_3, v_5]$	$[v_6, v_9, v_4]$
6	$v_6$	$[v_0, v_1, v_2, v_3, v_5, v_6]$	$[v_9, v_4, v_8]$
7	$v_9$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9]$	$[v_4, v_8]$
8	$v_4$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4]$	$[v_8]$
9	$v_8$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4, v_8]$	$[v_7]$
10	$v_7$	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4, v_8, v_7]$	$\emptyset$