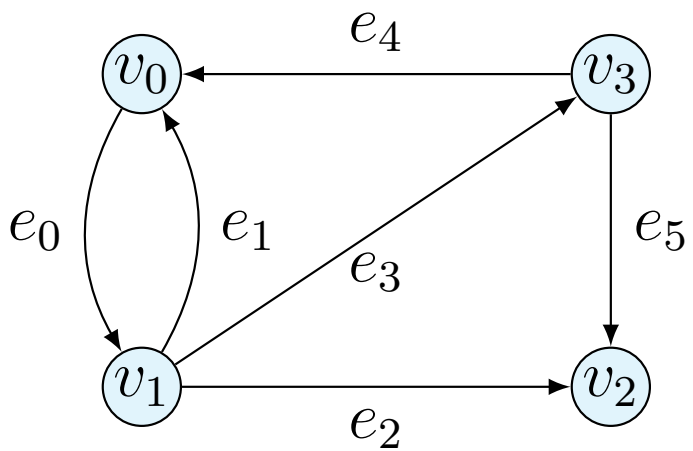
グラフの探索

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

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

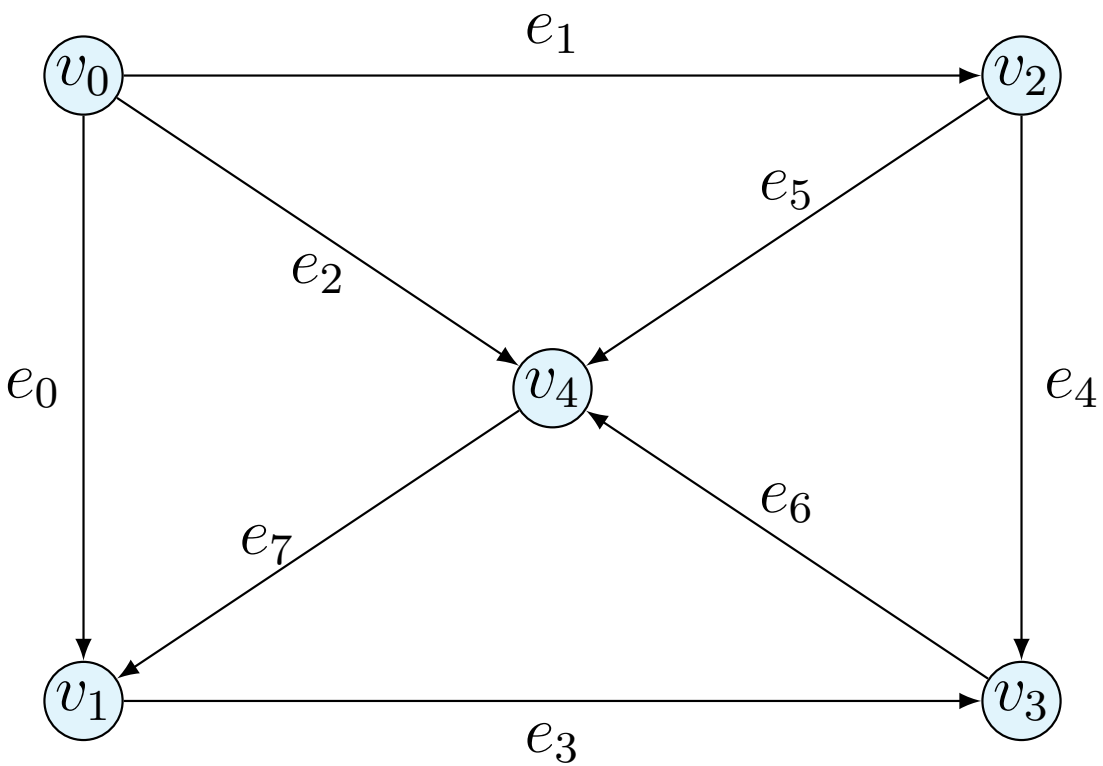


適切なアルゴリズムを作る

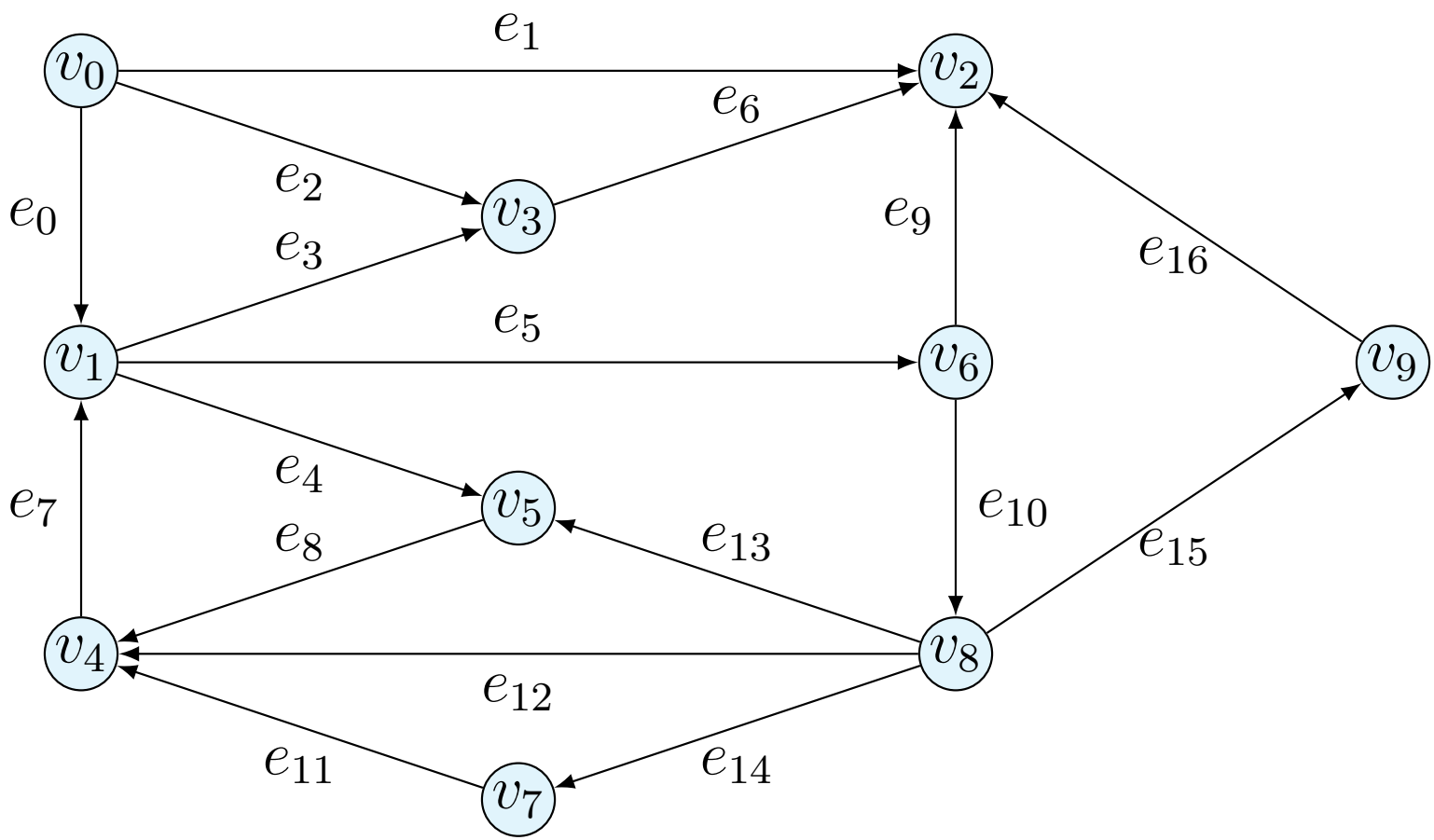
深さ優先探索 DFS: Depth-First Search

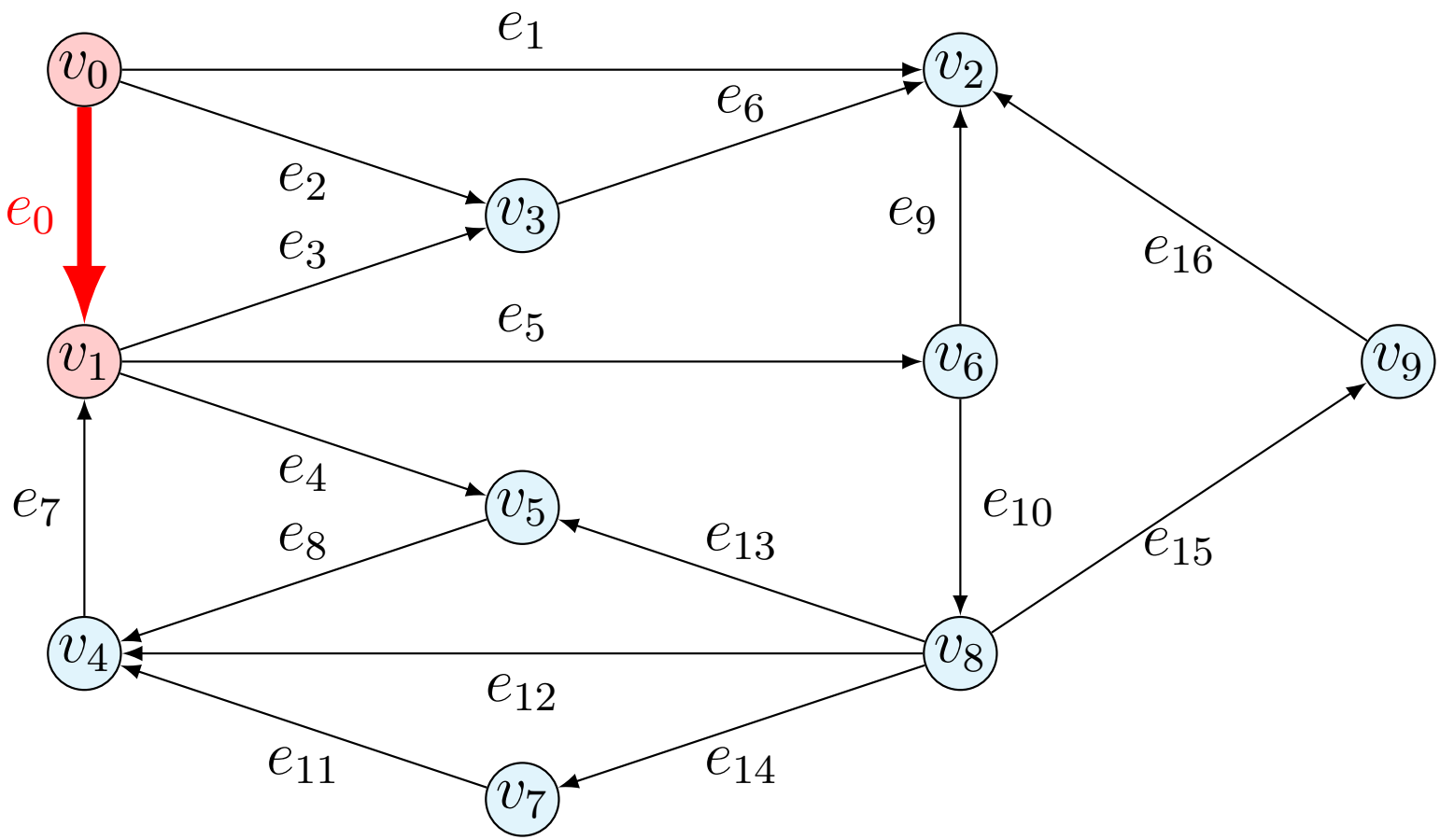
- 出発点を定める
- たどれる限り、辺をたどる: 再帰的アルゴリズム
  - それ以上進めなくなるまで
  - 新たな点が無くなるまで
- 道を戻って、別の辺をたどる
- 結果としてできる木 (spanning tree) は、深いものができる

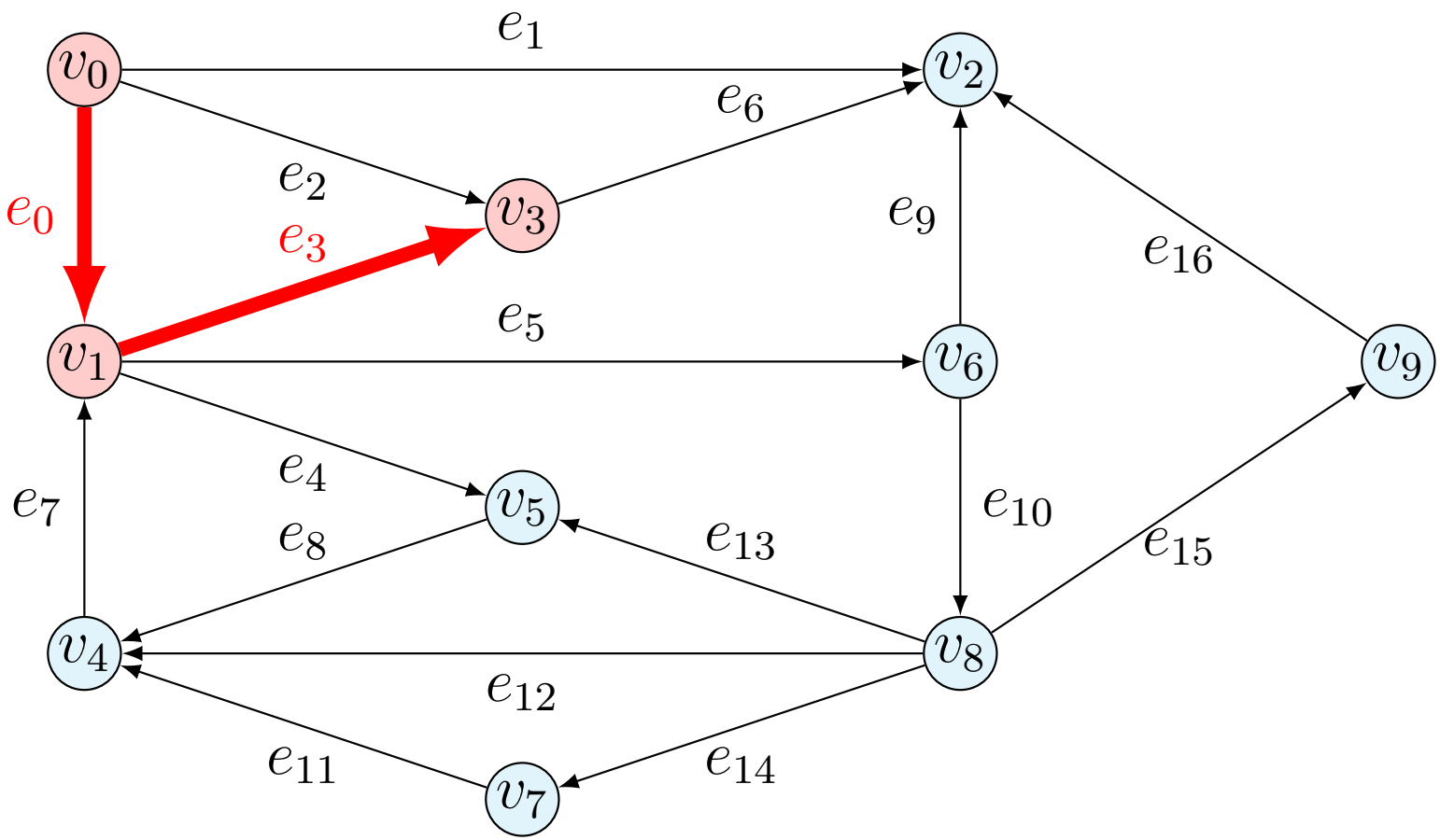
例 1.1:

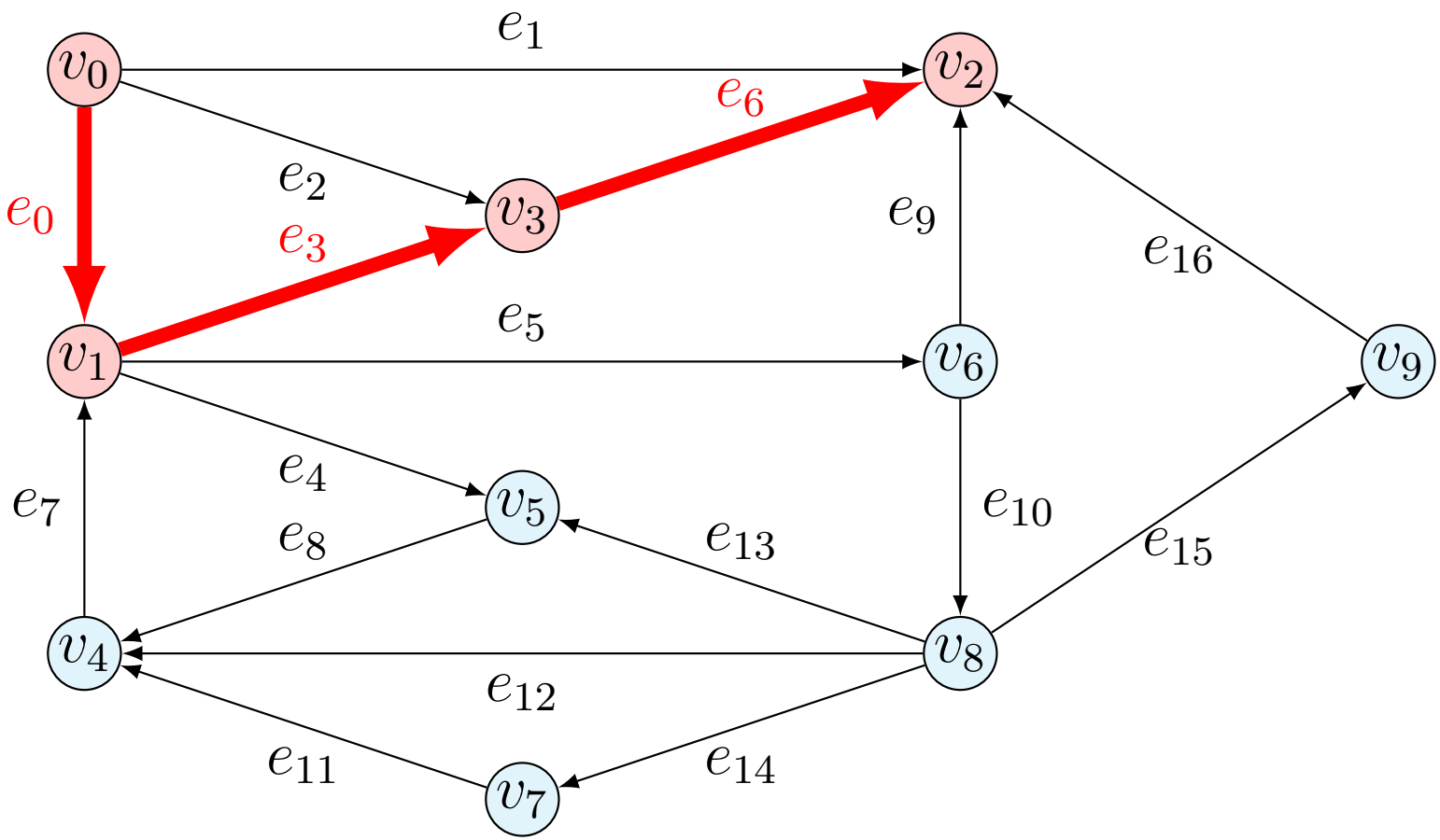


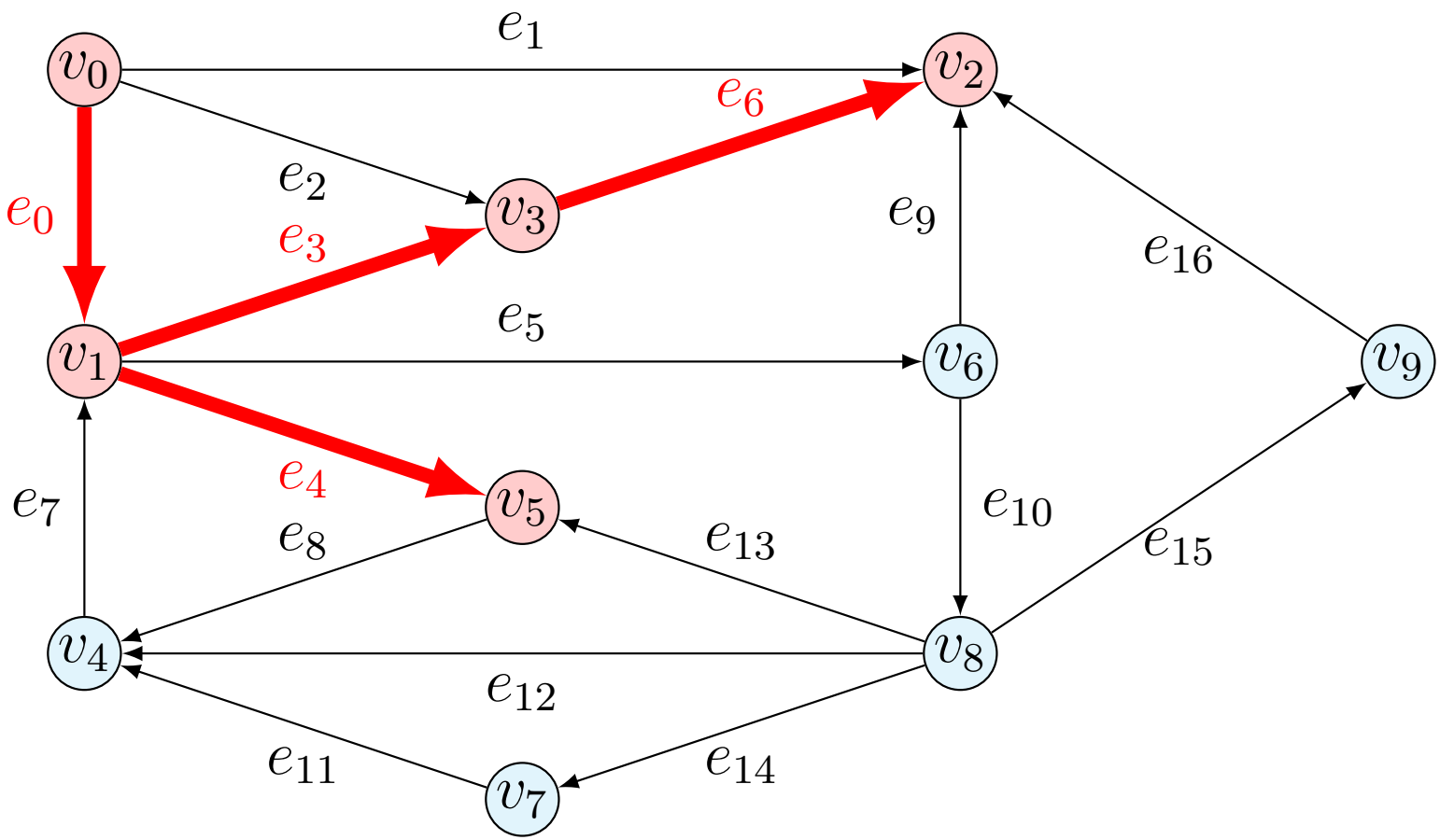
例 1.2:

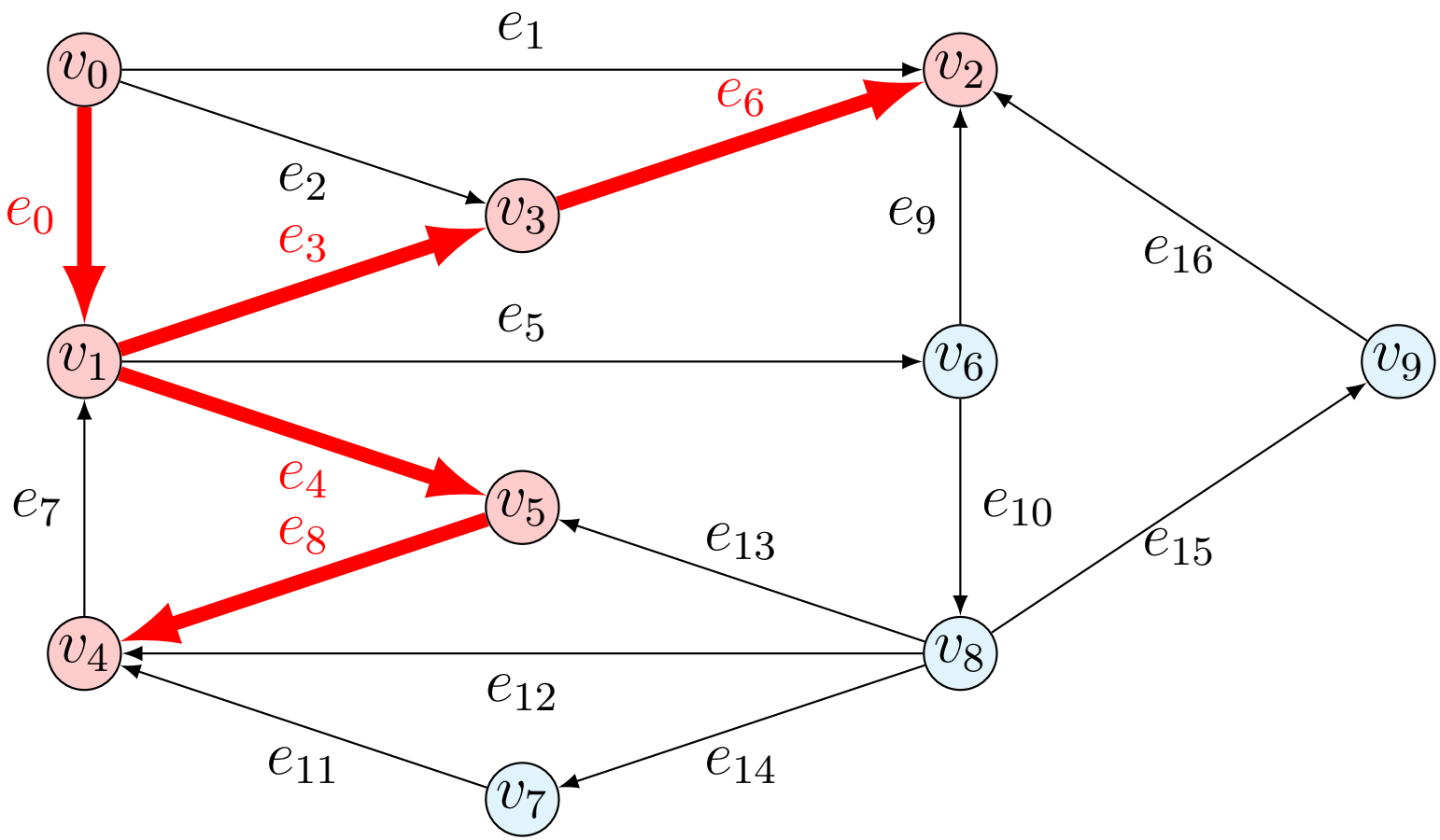

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

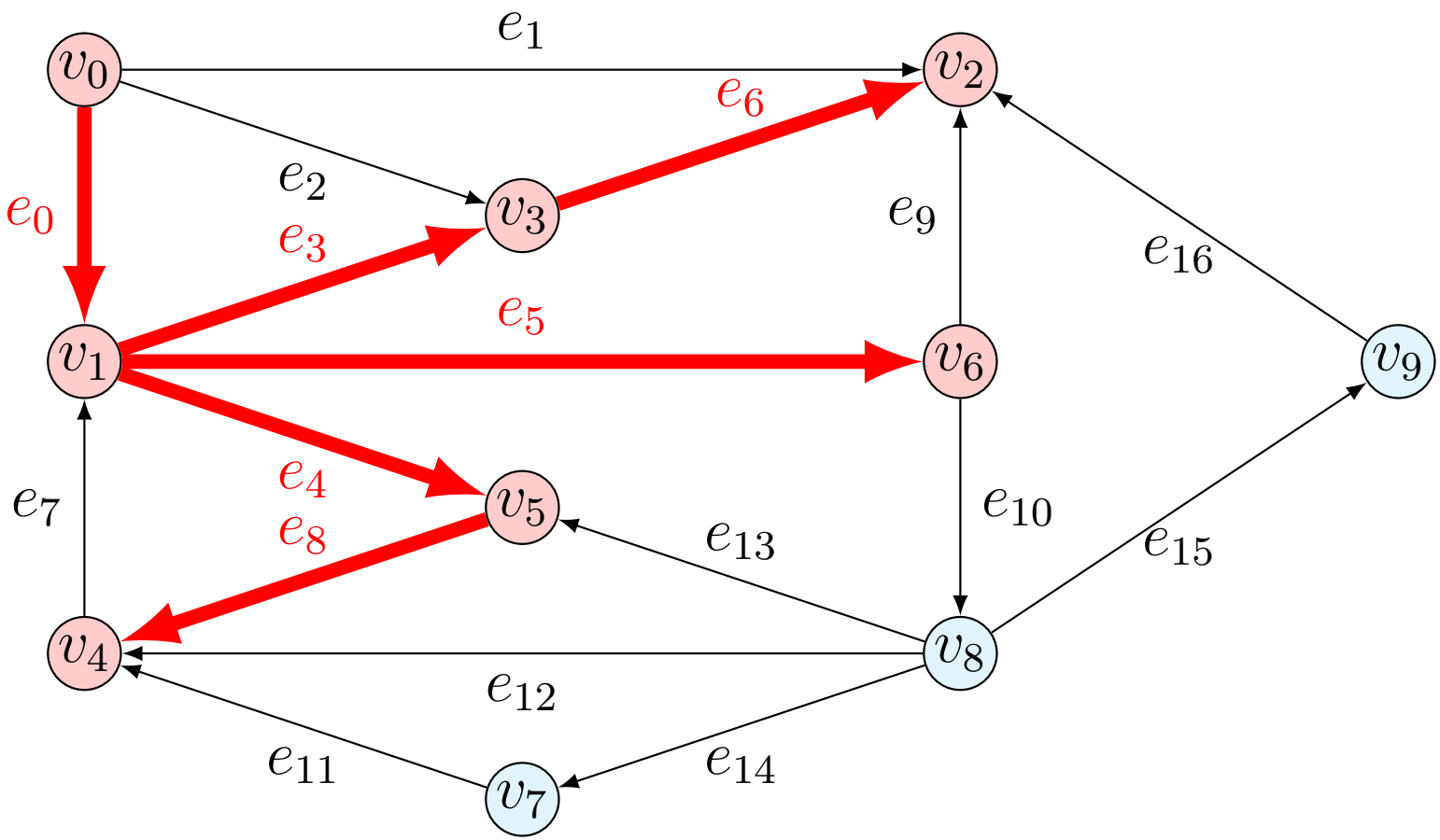

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

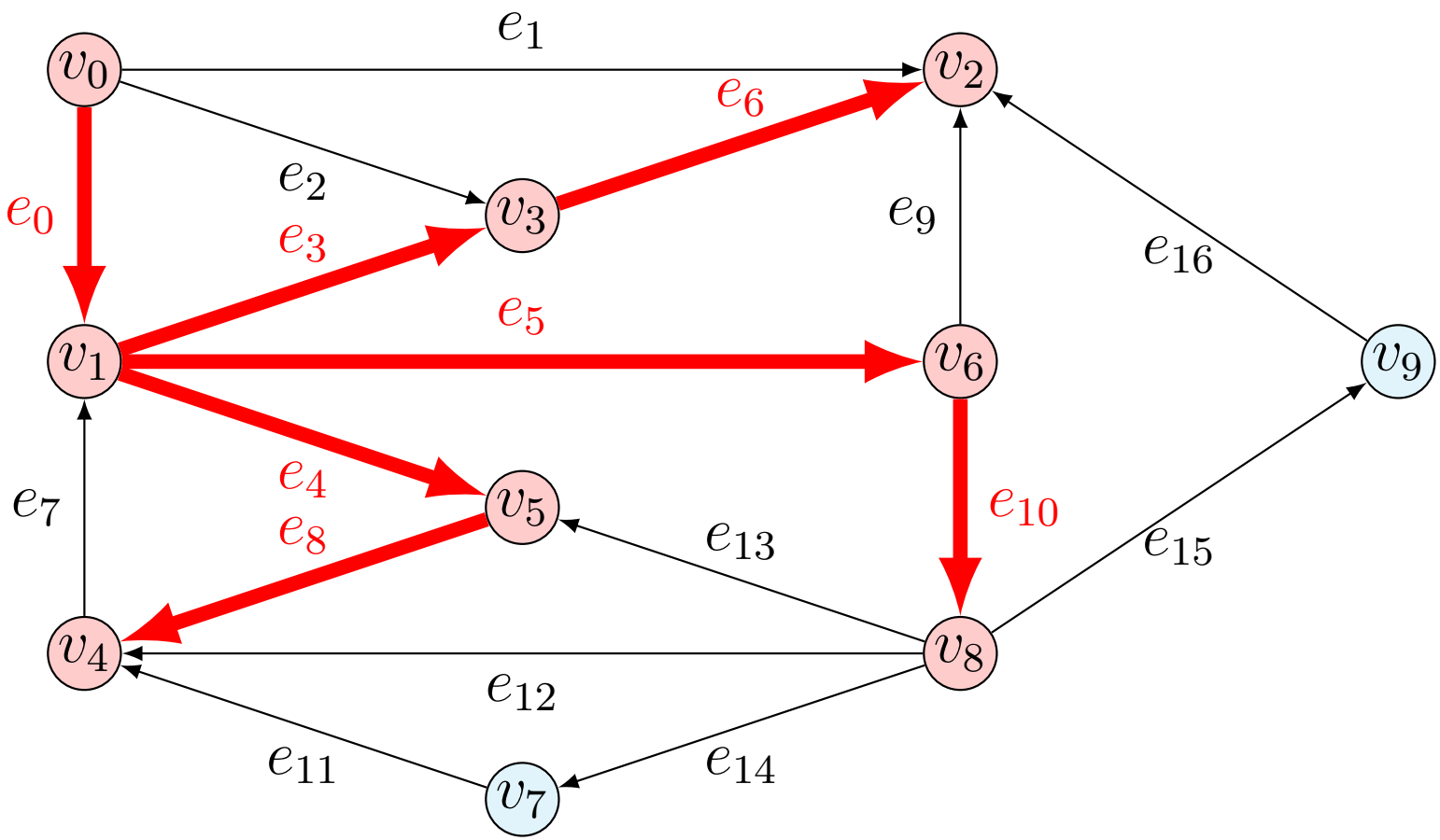

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

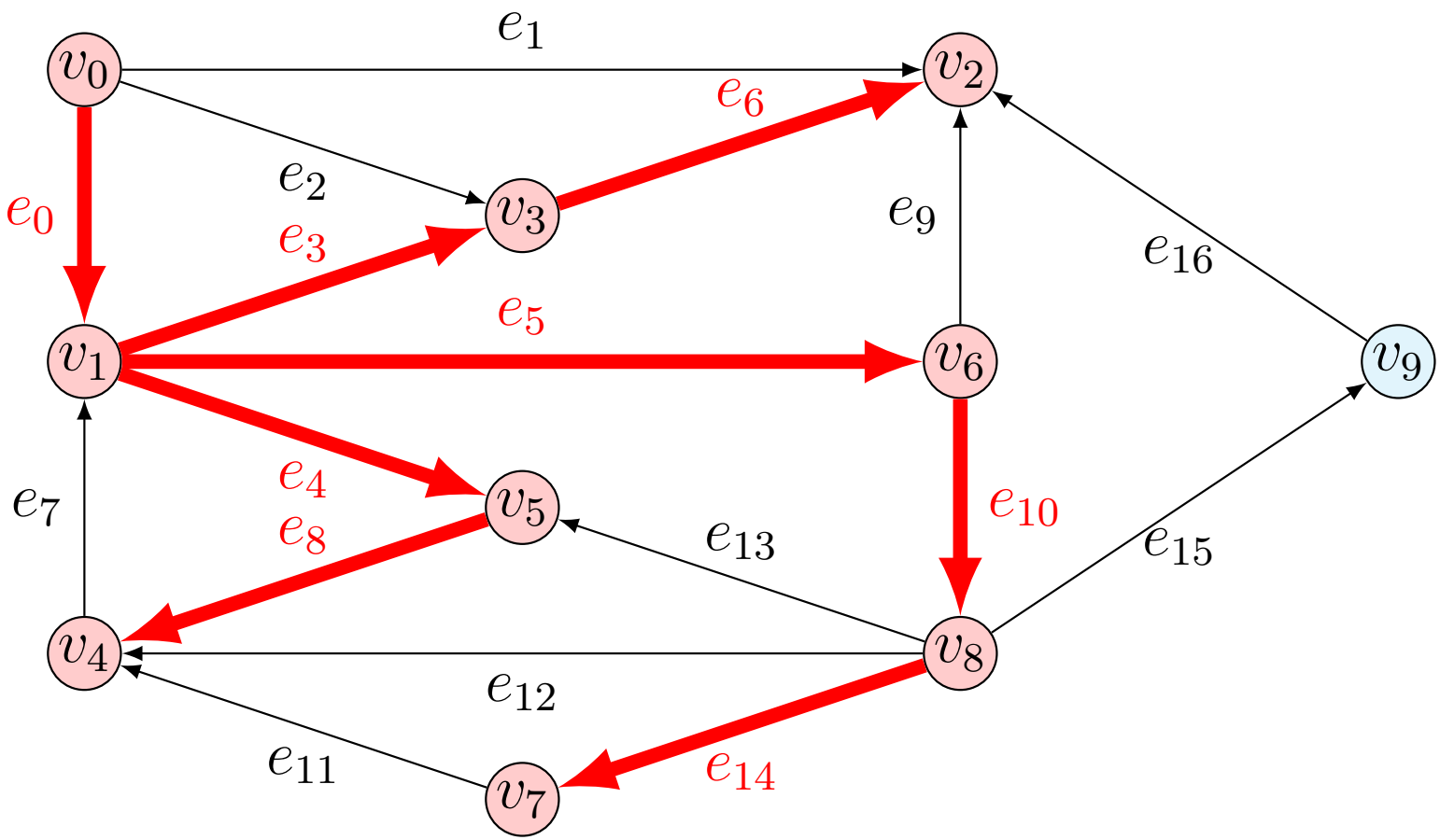

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

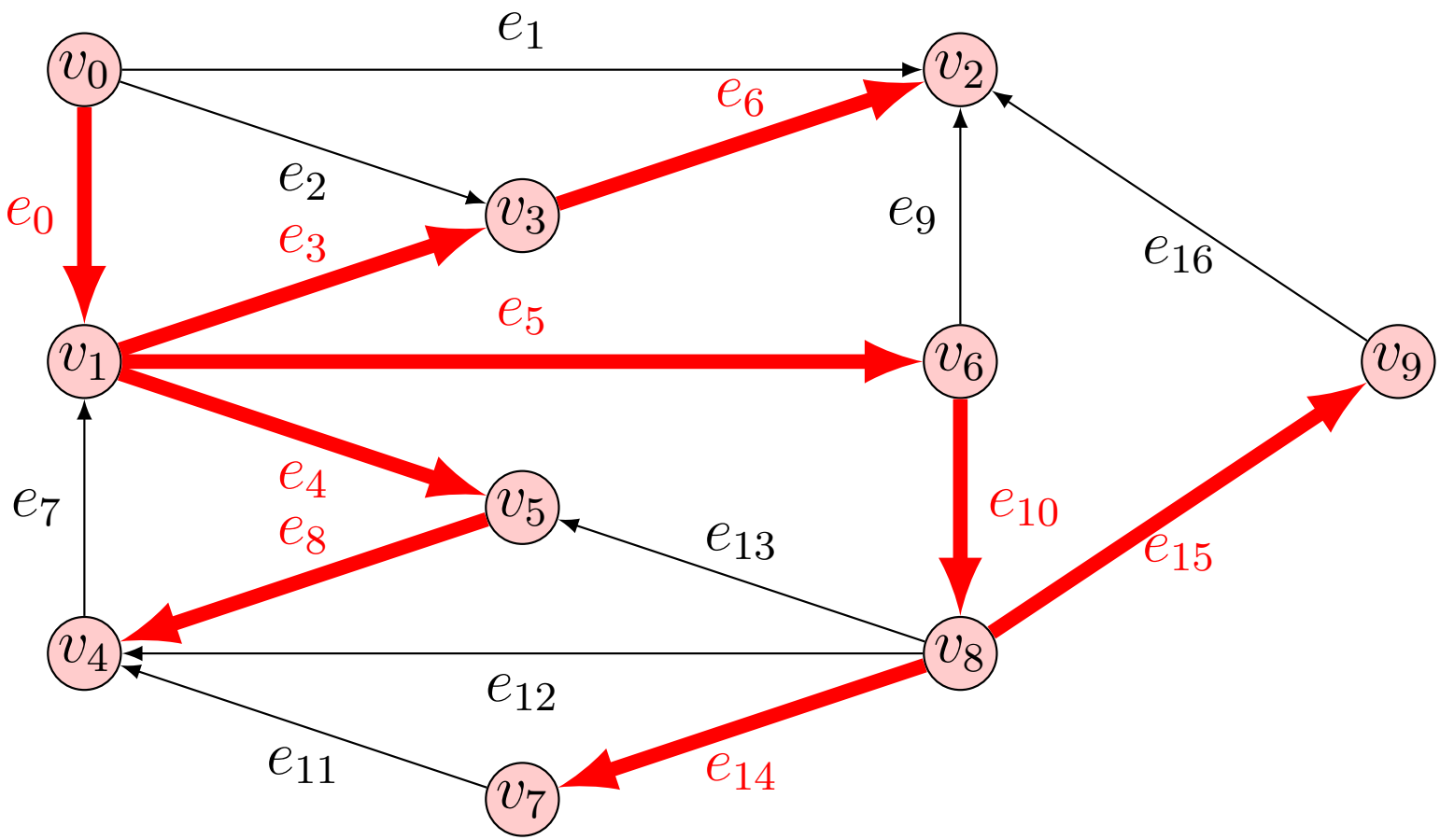

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

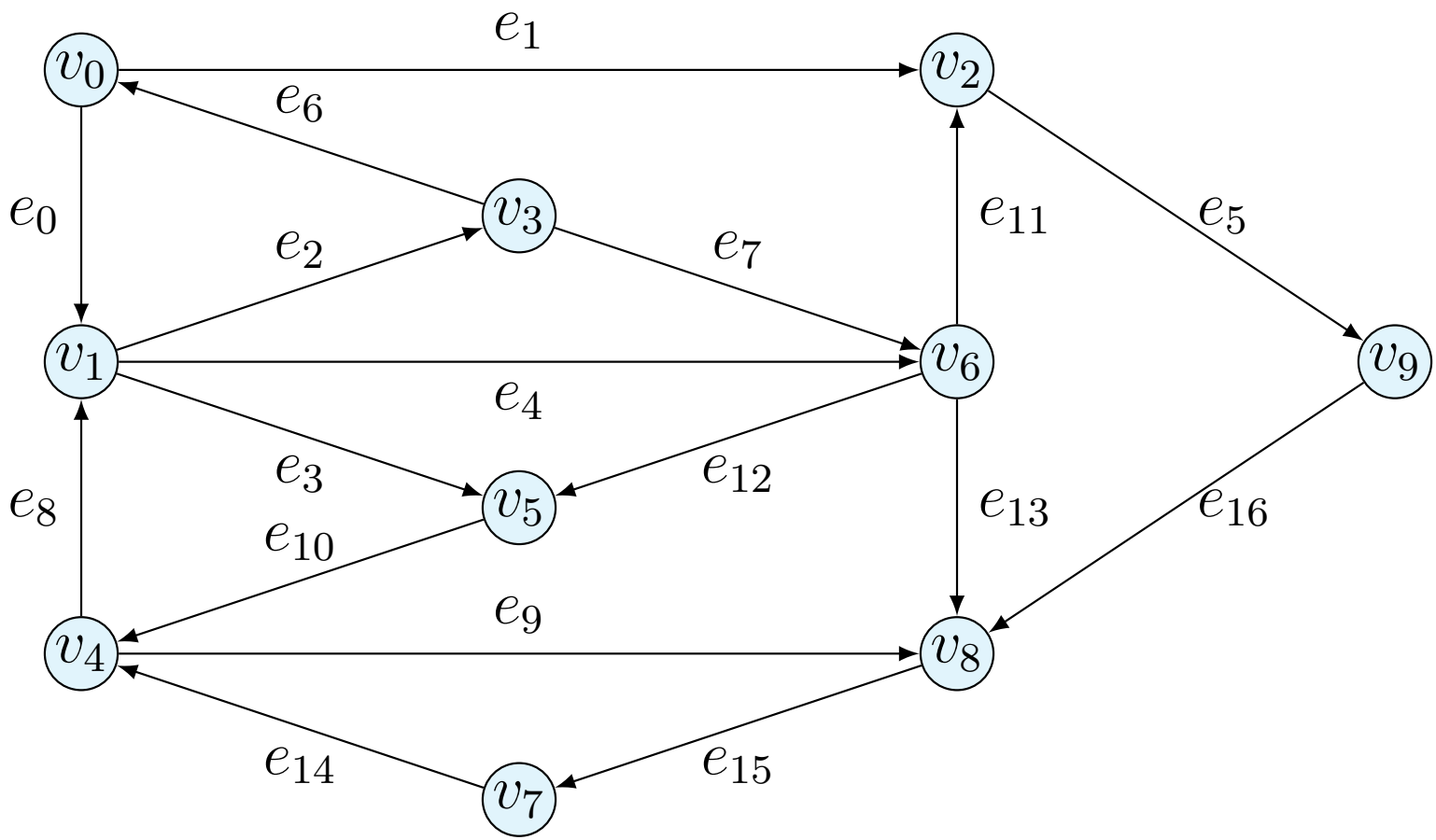
Algorithm 1 DFA アルゴリズム

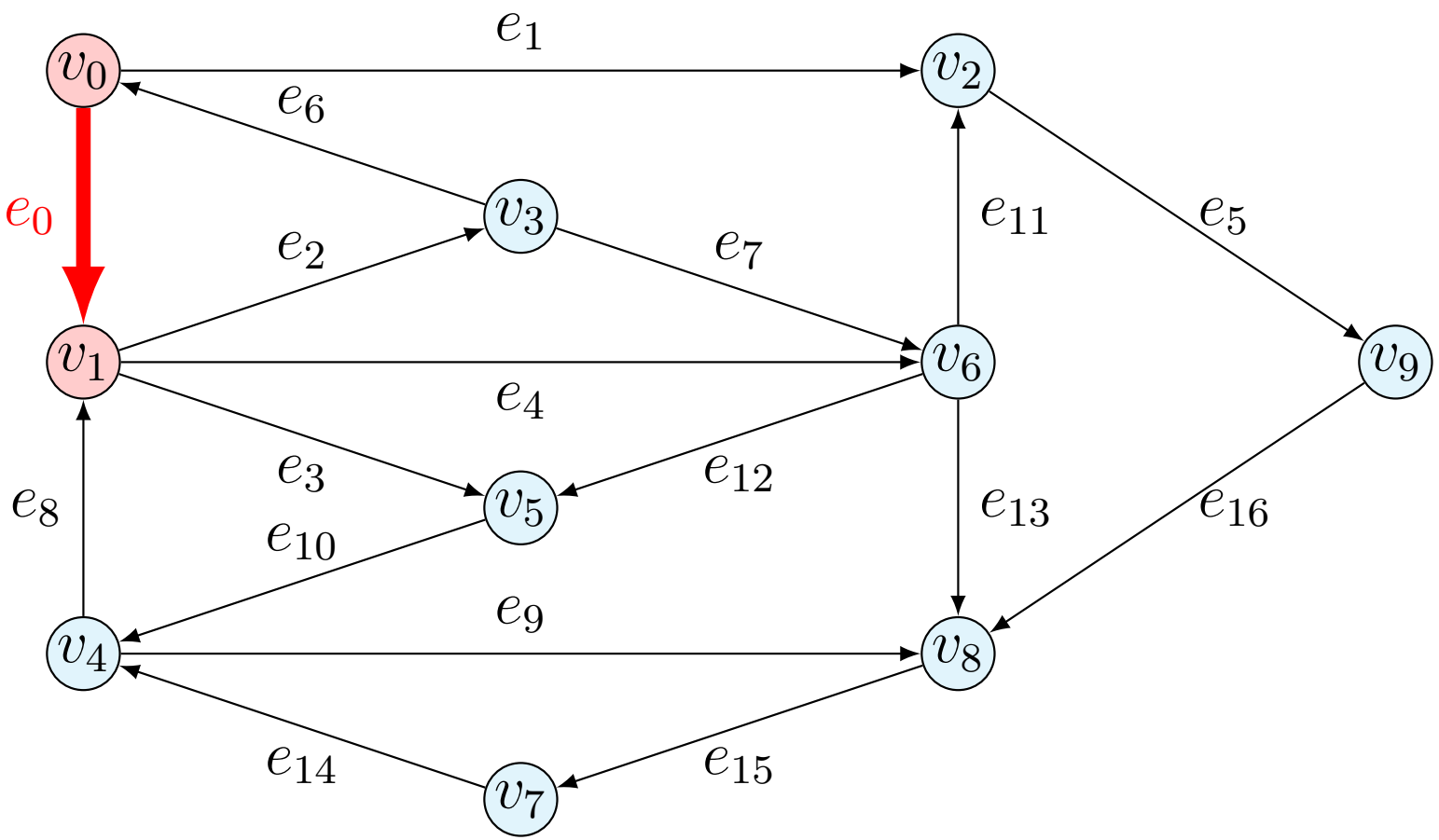
procedure $\textsc{search}(v, L)$
  for all $e \in \delta^+ v$ **do** ▷ $//v$ から出る全ての辺
    $w = \partial^- e$
    if $w \notin L$ **then**
      L.append(w)
      $\textsc{search}(w, L)$
    end if
  end for
end procedure

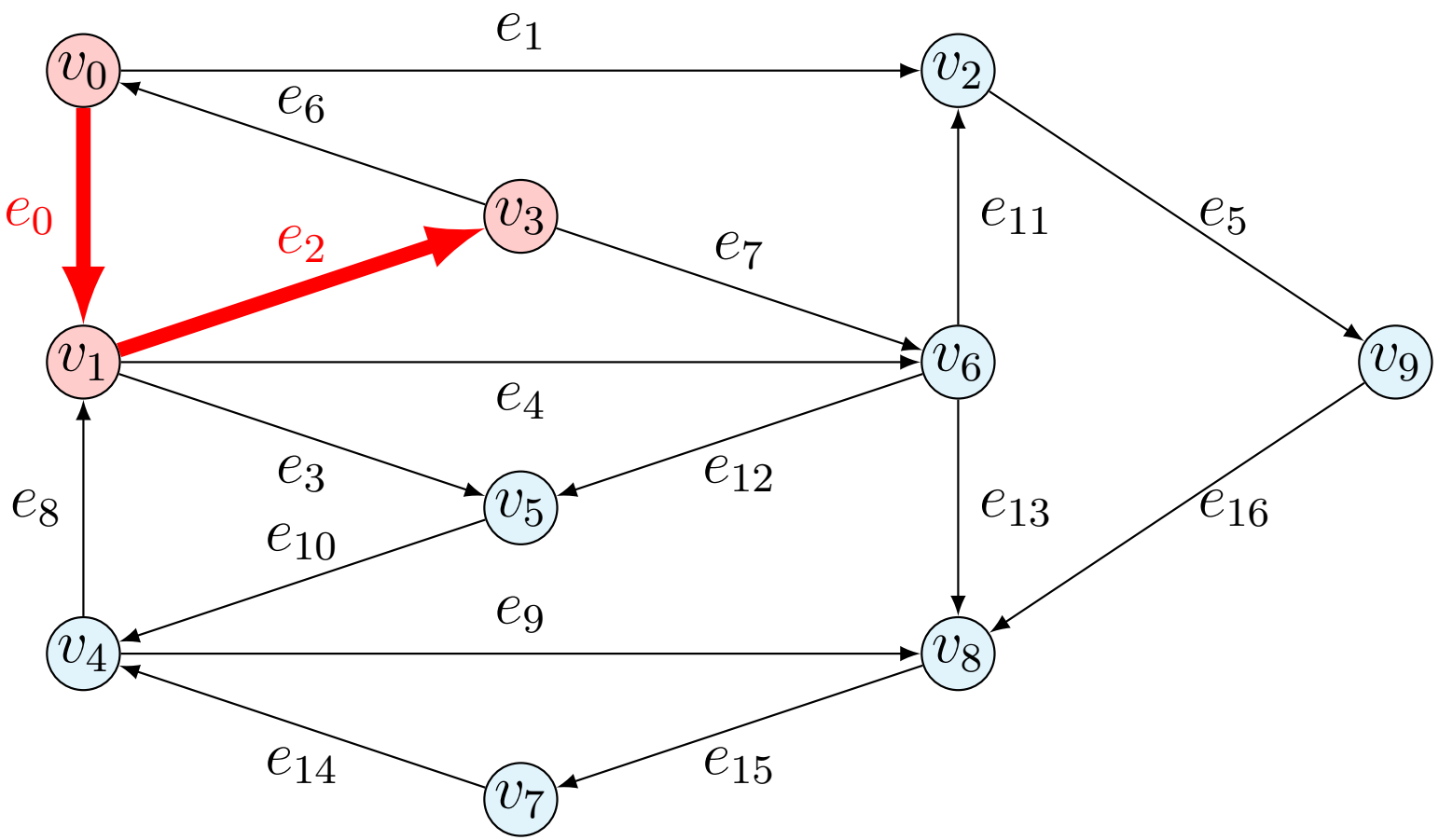
実行状況

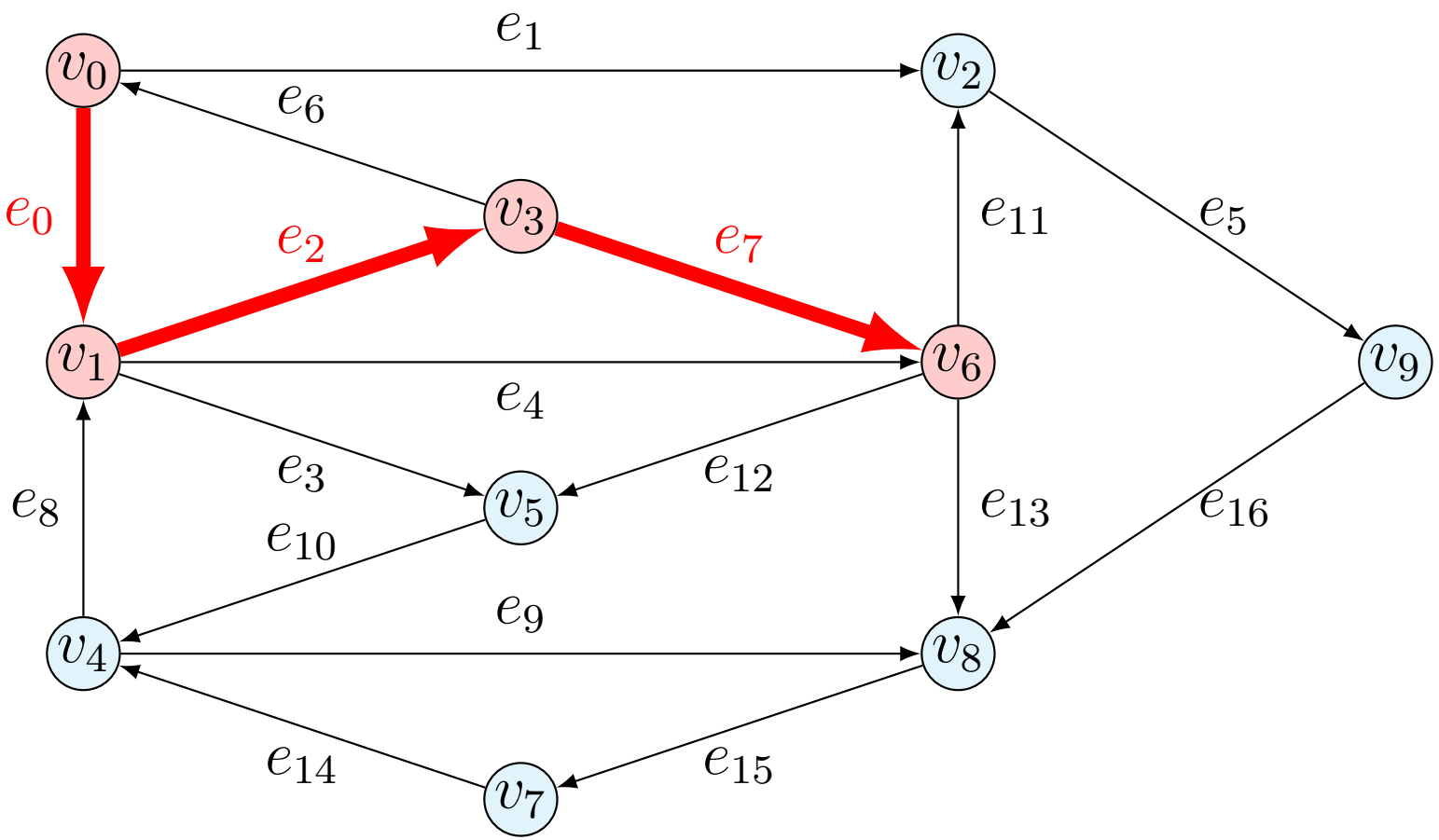
$$
\begin{aligned}
(v_0, [v_0]) &\to (v_1, [v_0, v_1]) \\
&\to (v_3, [v_0, v_1, v_3]) \\
&\to (v_2, [v_0, v_1, v_3, v_2]) \\
&\to (v_5, [v_0, v_1, v_3, v_2, v_5]) \\
&\to (v_4, [v_0, v_1, v_3, v_2, v_5, v_4]) \\
&\to (v_6, [v_0, v_1, v_3, v_2, v_5, v_4, v_6]) \\
&\to (v_8, [v_0, v_1, v_3, v_2, v_5, v_4,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

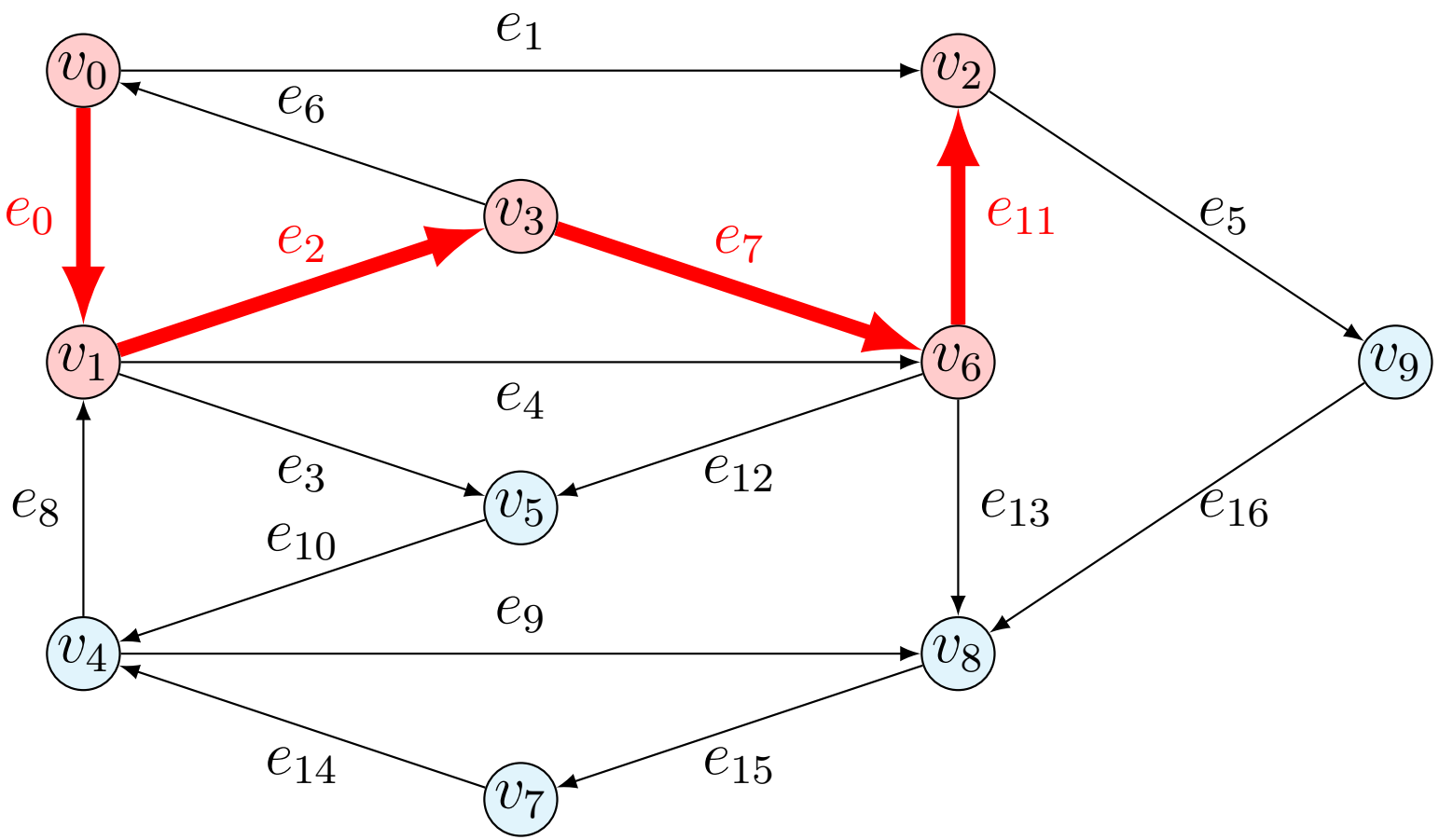
例 1.3:

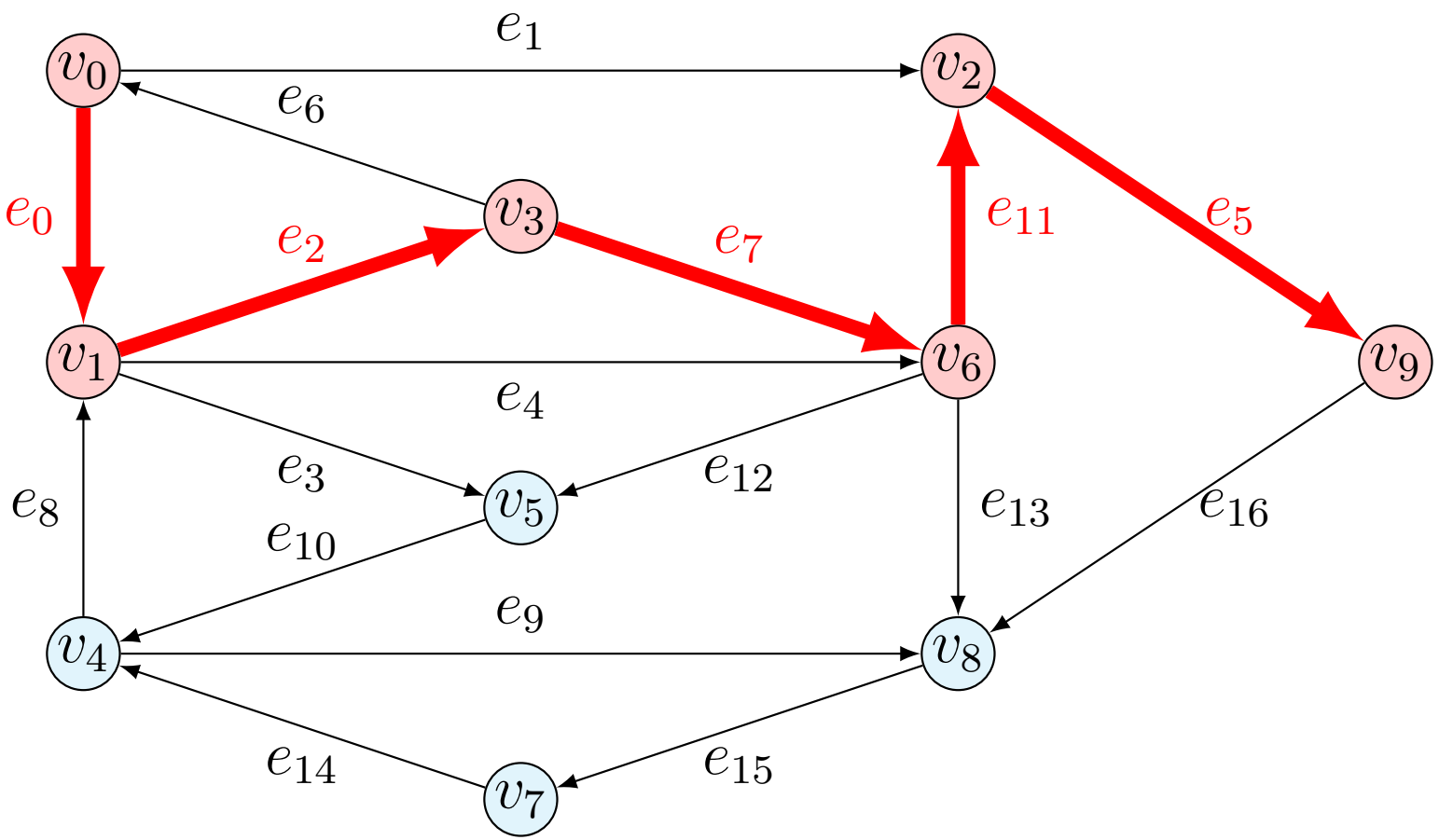

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

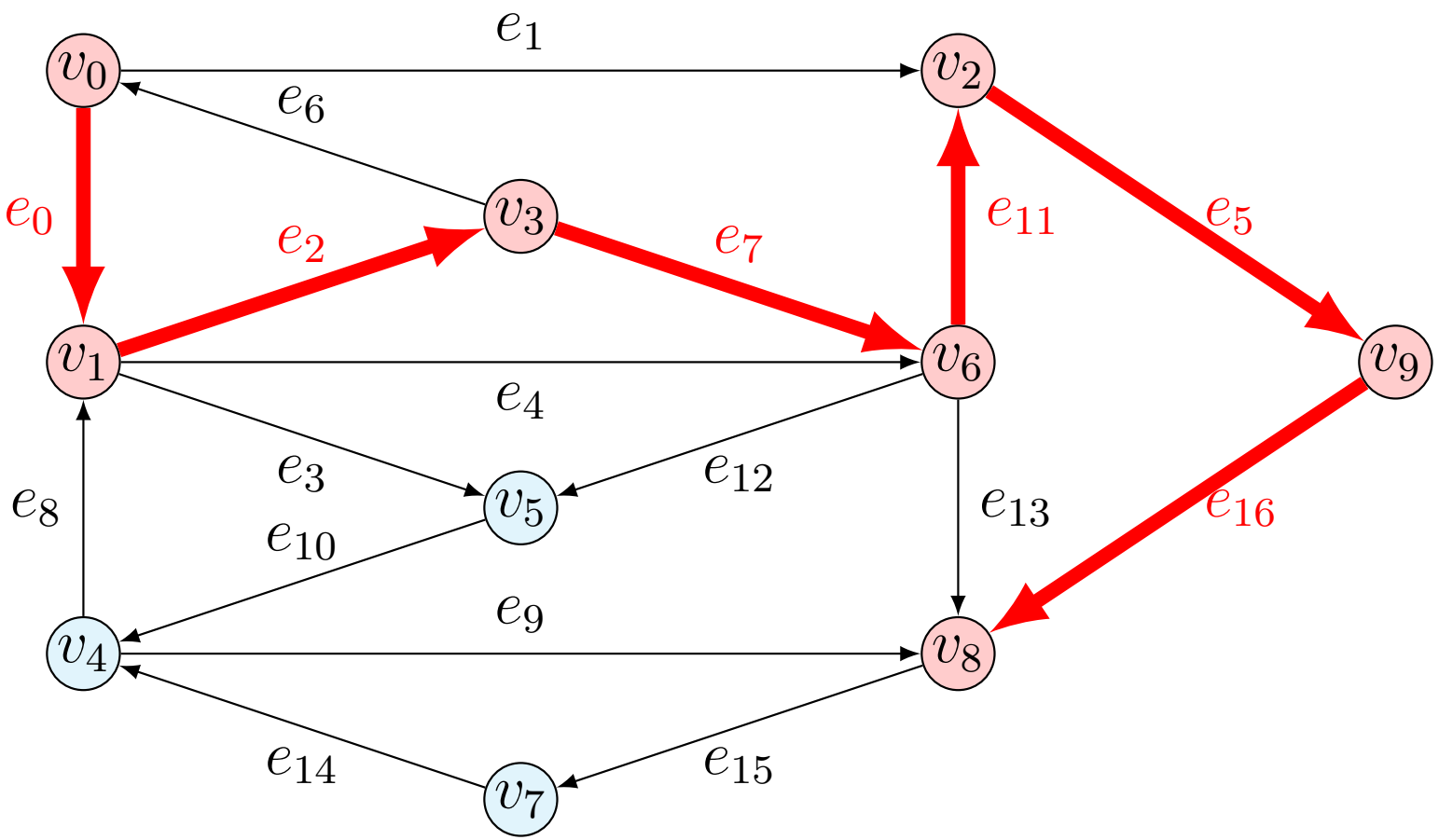

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

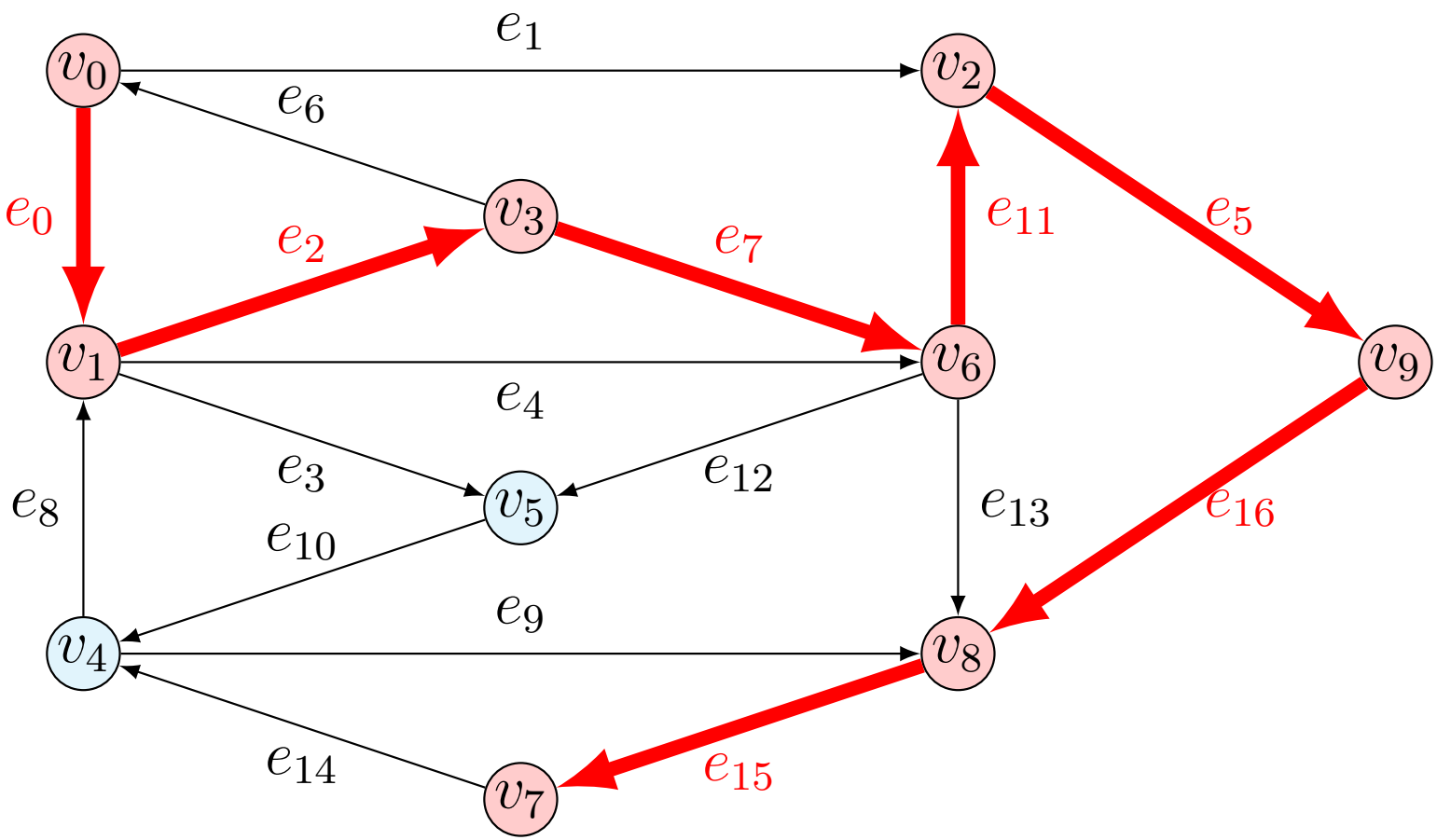

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

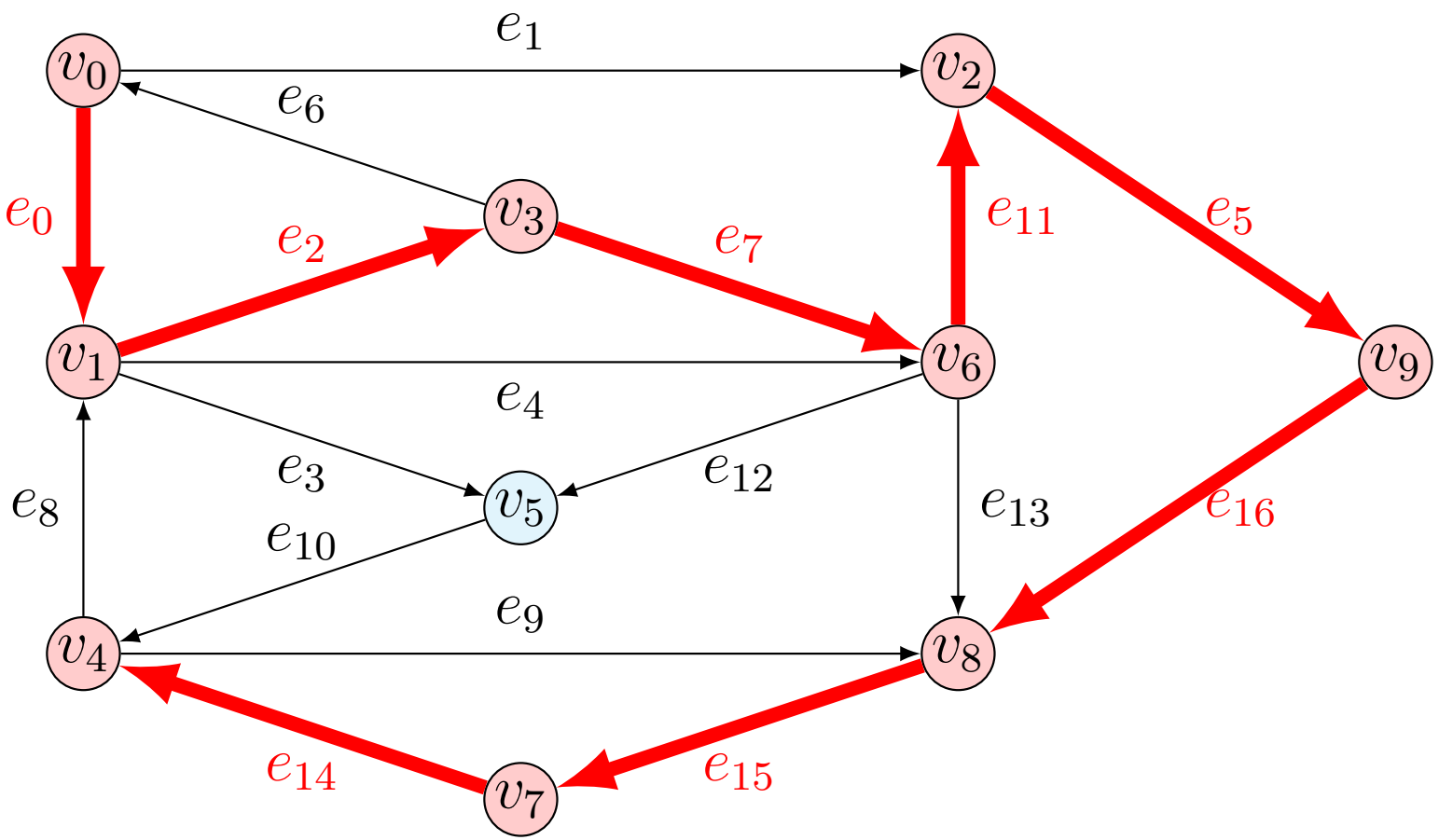

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

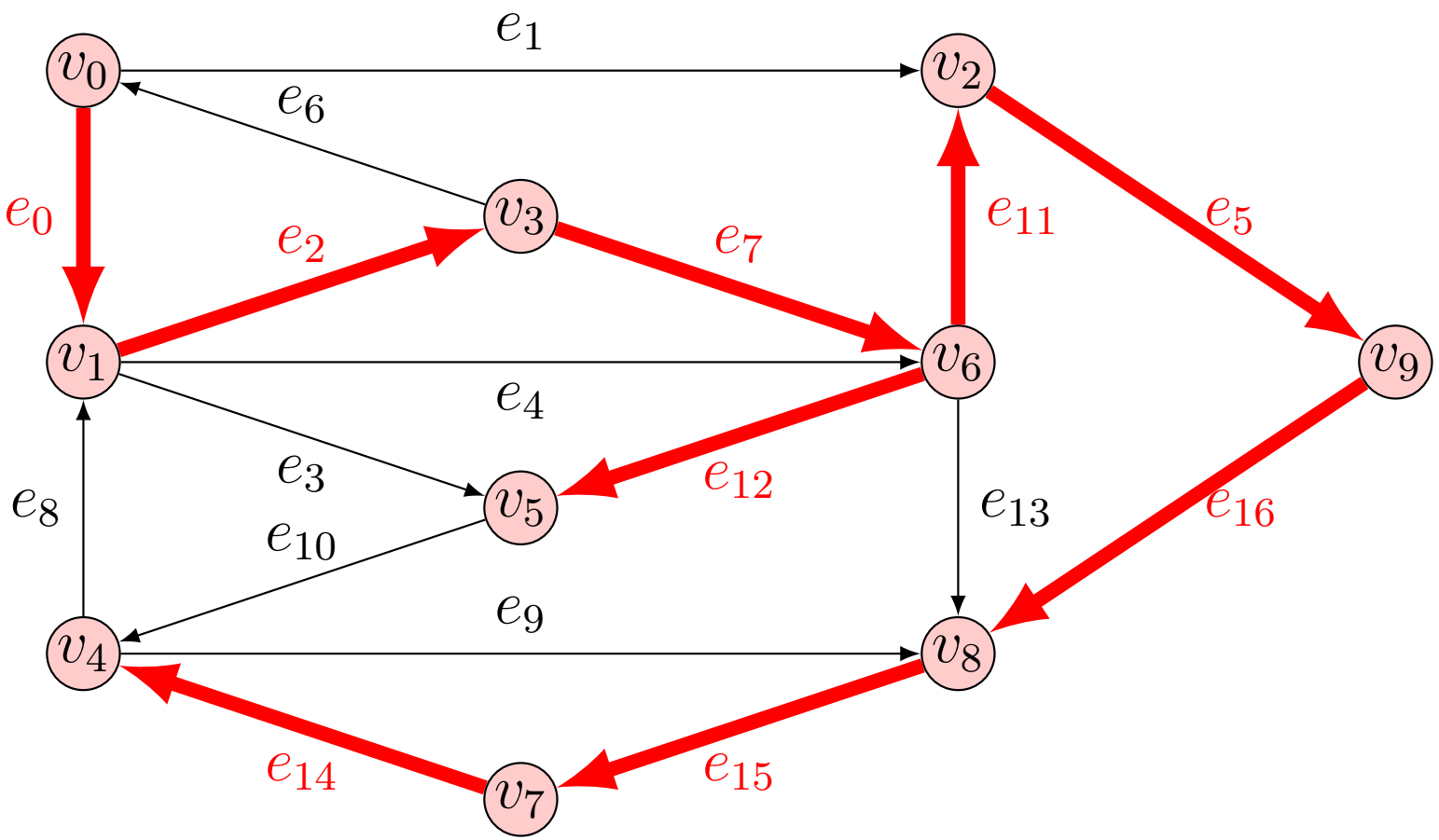

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

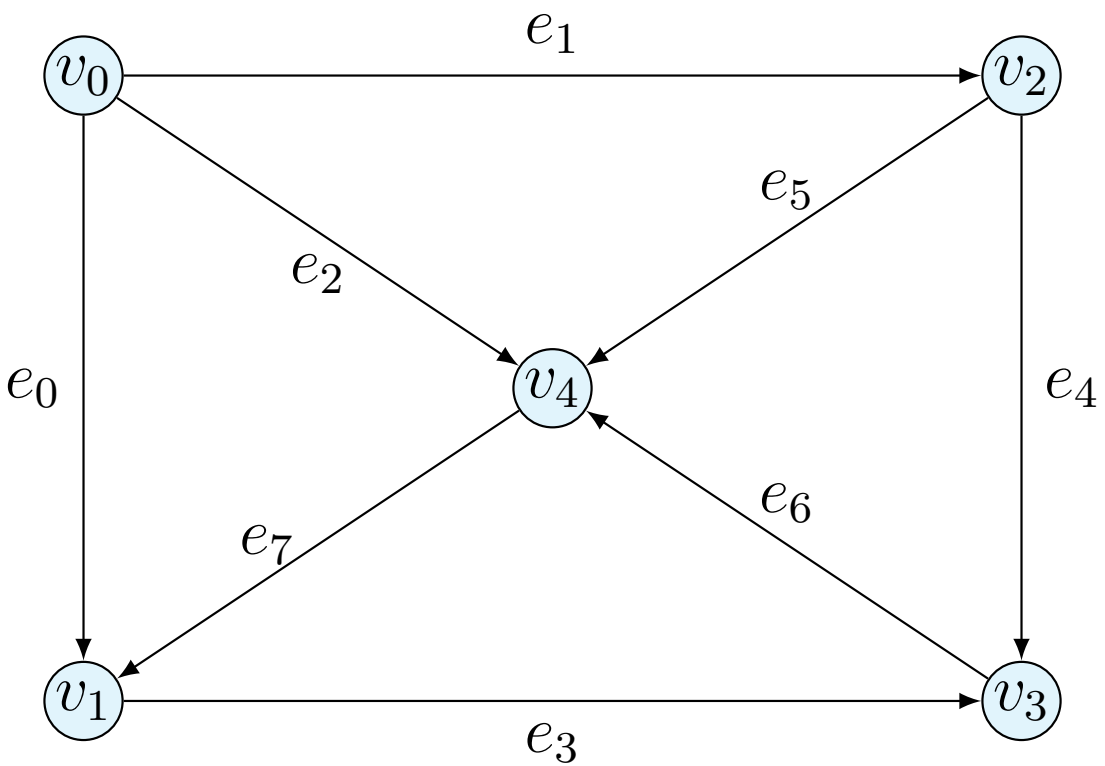

v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

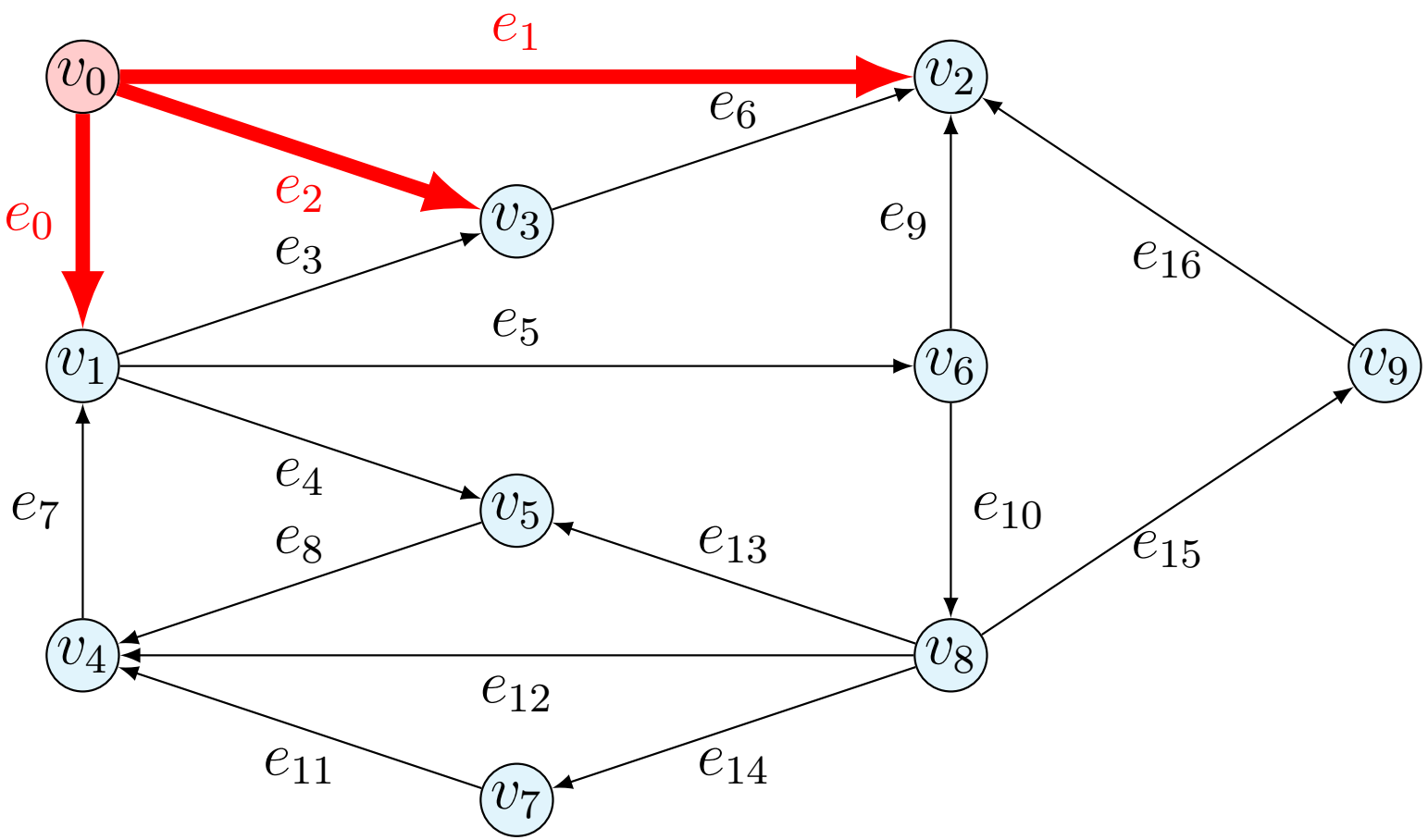
幅優先探索 BFS: Dreadth-First Search

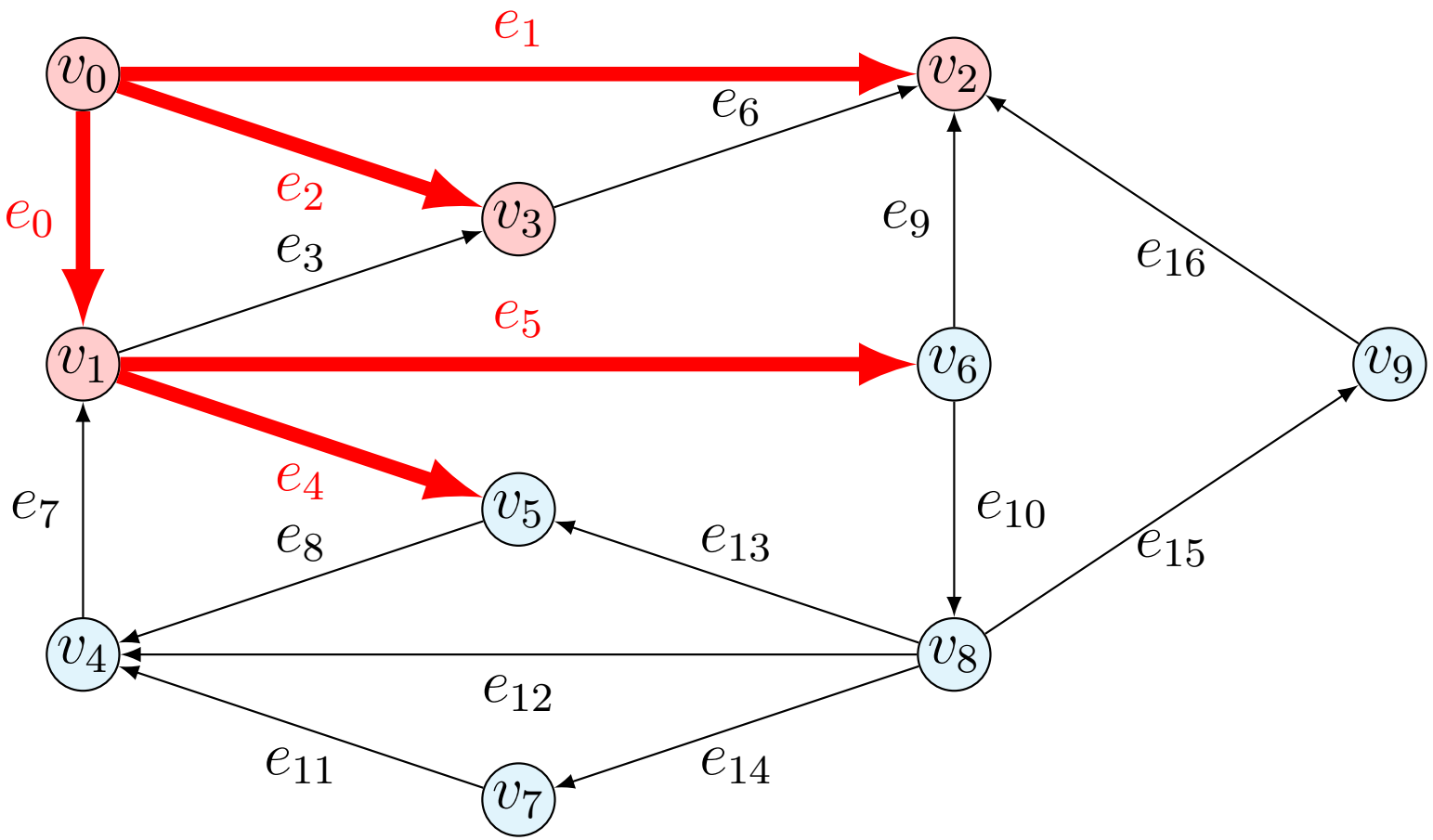
- 出発点を定める。この点の集合を S_0 とする。
- 新たな頂点がなくなるまで繰り返す
  - S_{i-1} の各点の隣接頂点のうち、未調査の点の集合を S_i とする
- 結果としてできる木 (spanning tree) は、浅いものができる

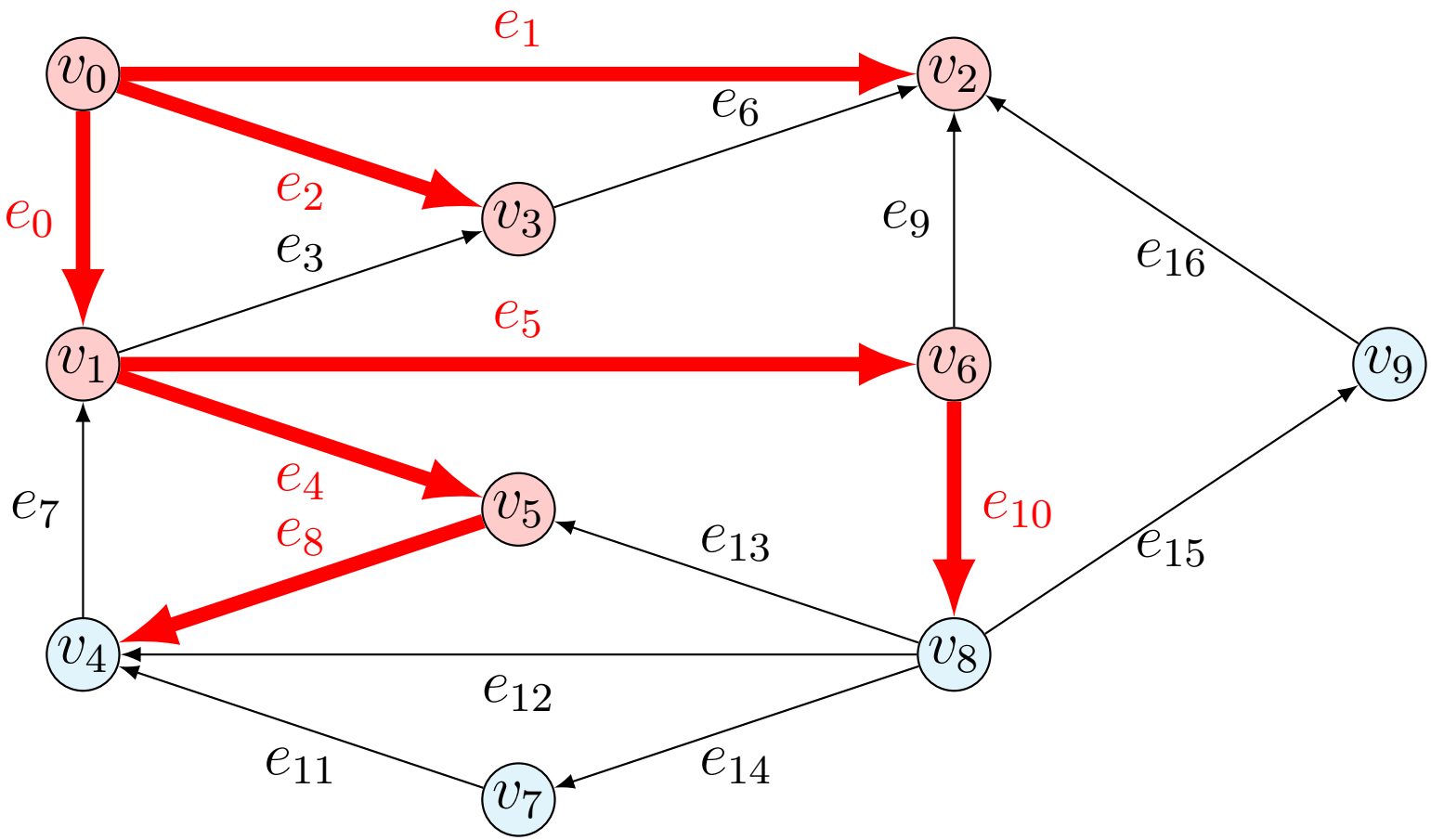
例 2.1:

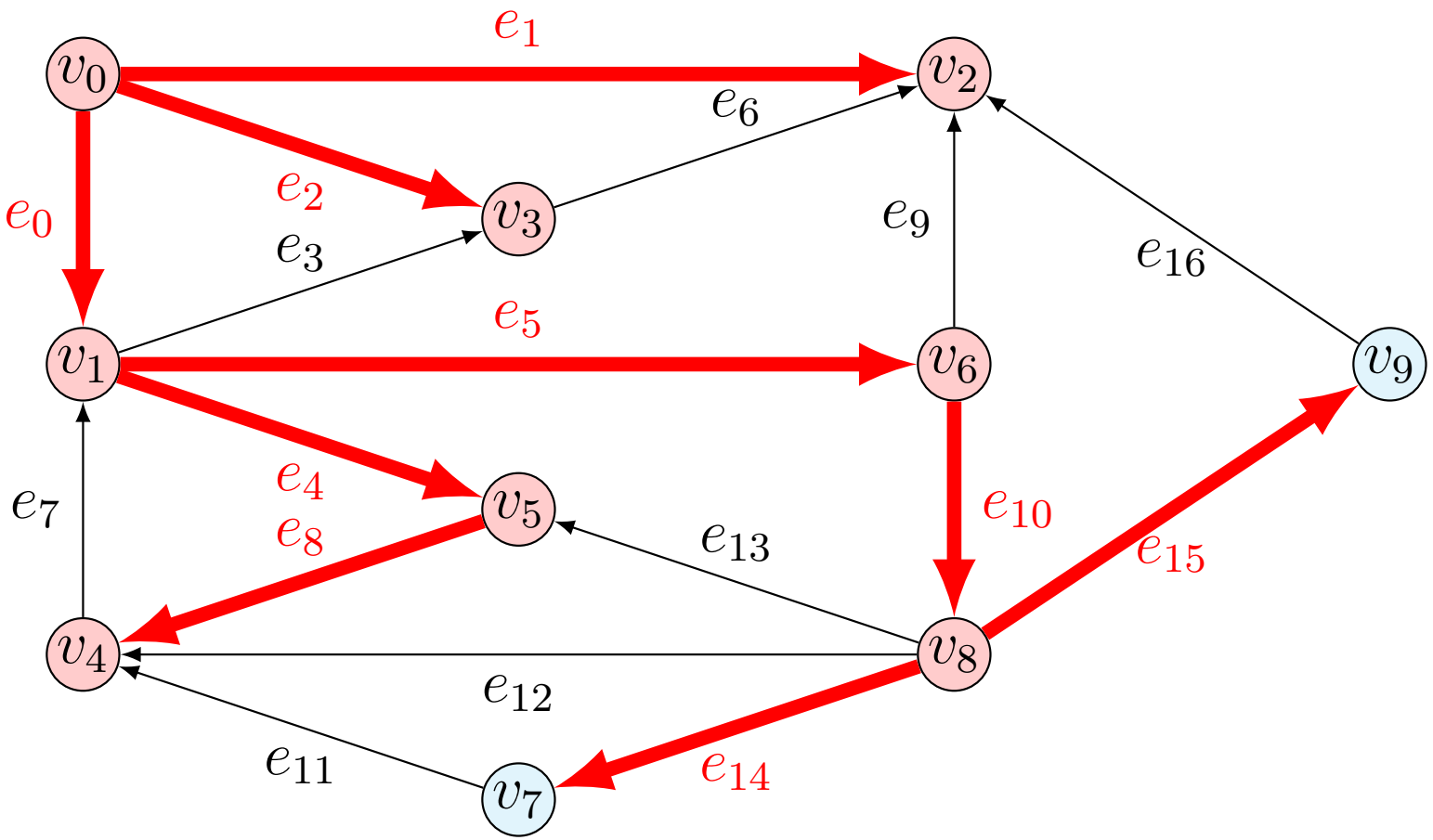


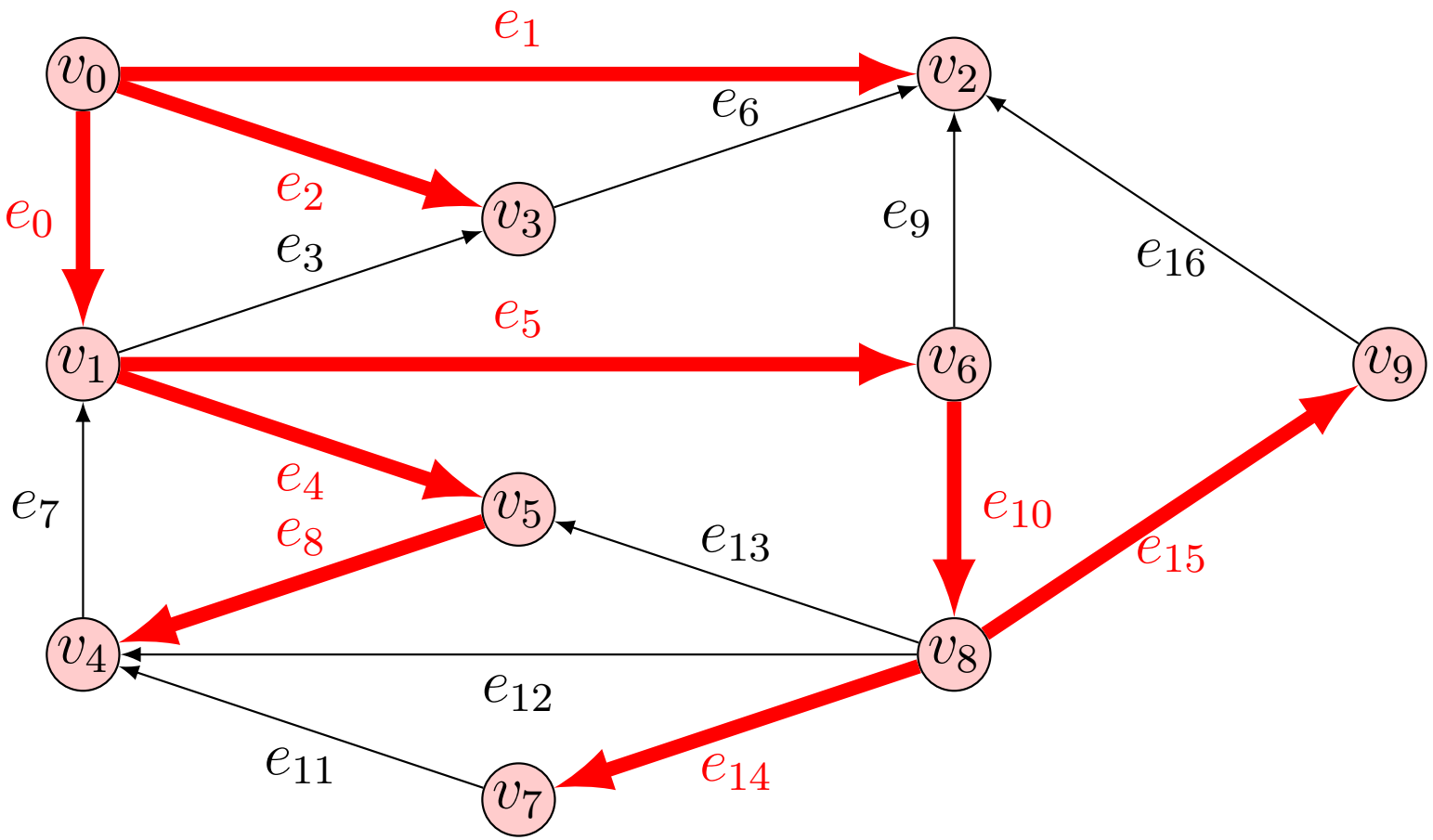
例 1.2 に対する BFS の結果




v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

Algorithm 2 BFS アルゴリズム

$L = \emptyset$ ▷ 到達済み頂点のリスト
$Q = [r]$ ▷ 調査すべき頂点の待ち行列
while $Q \neq \emptyset$ **do**
  $v = Q$.poke() ▷ 待ち行列の先頭要素を取り出す
  for all $e \in \delta^+ v$ **do**
    $w = \partial^- e$
    if $w \notin L \wedge w \notin Q$ **then**
      Q.push(w)
    end if
  end for
  L.append(w)
end while

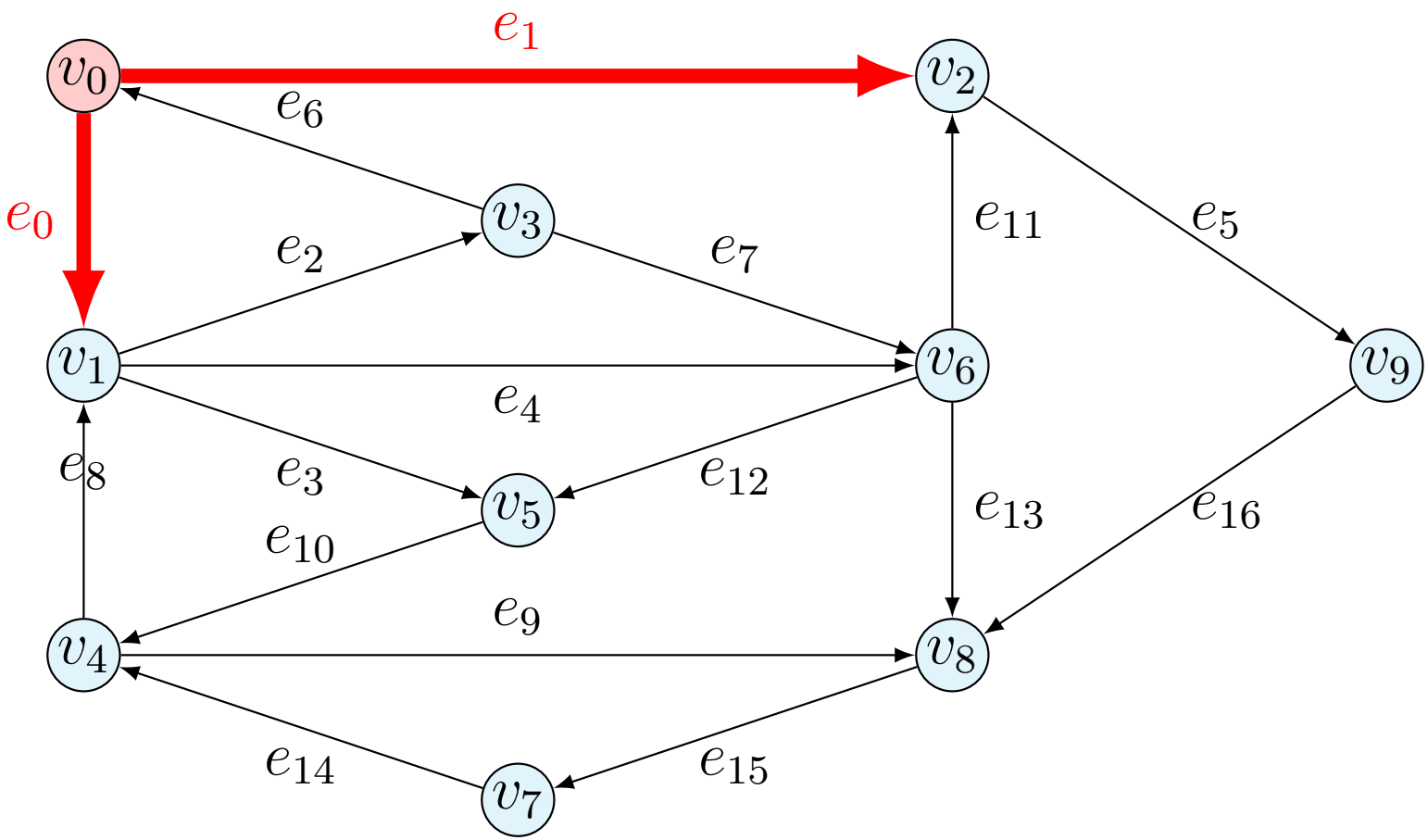
待ち行列: Queue

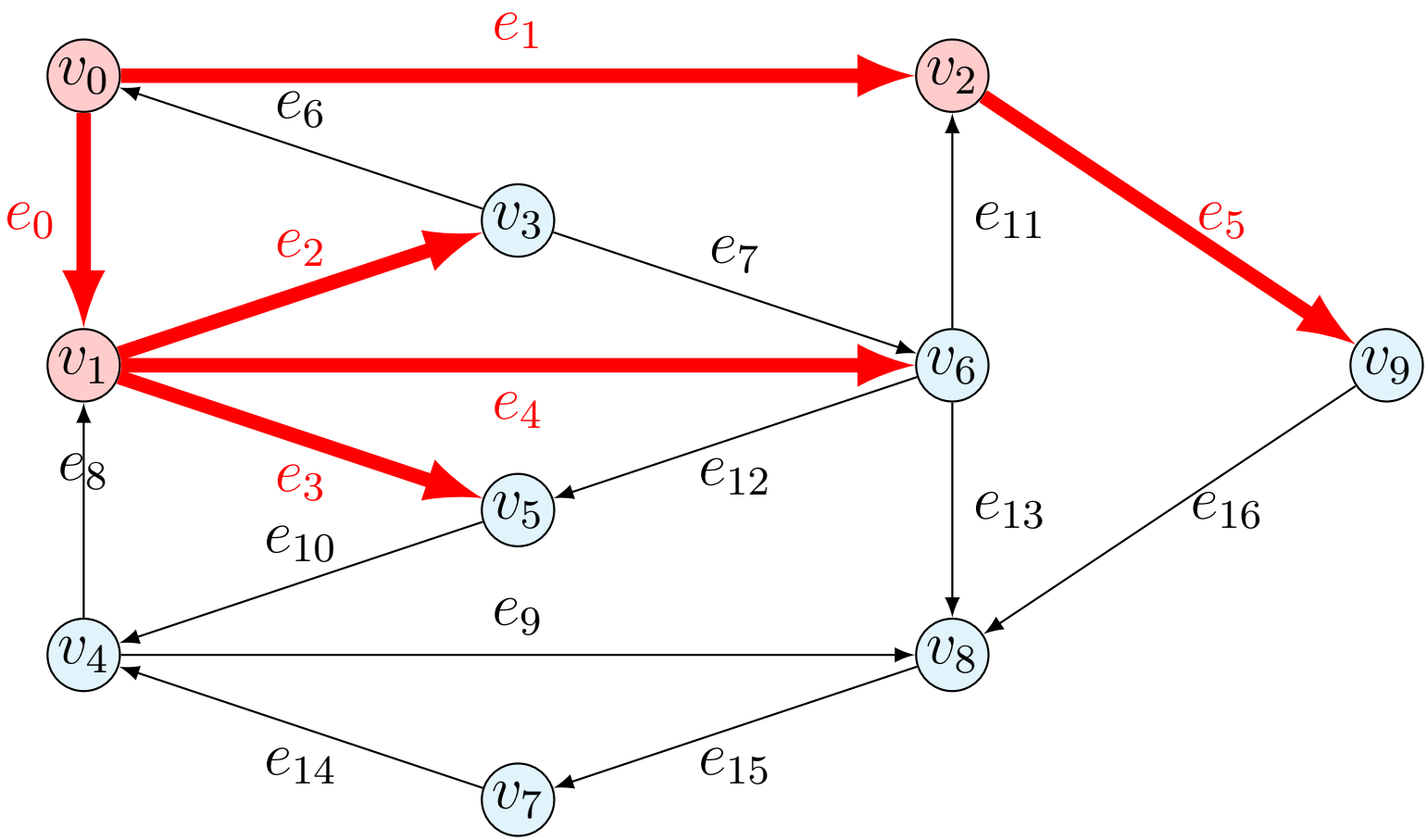
- リストの一種
- 末尾から要素を追加
- 先頭から要素を削除
- First-In-First-Out

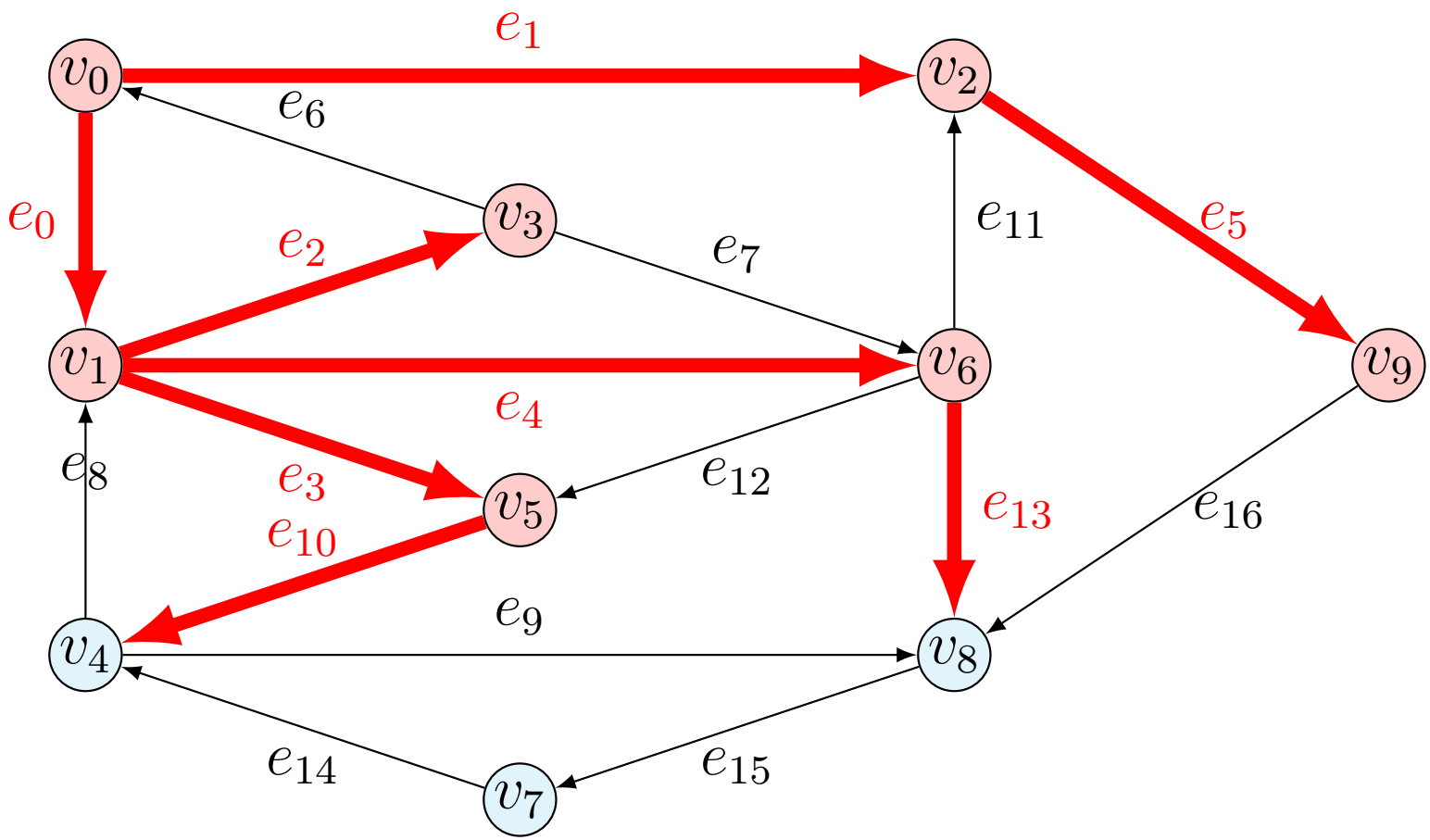
例 1.2: 探索の状況

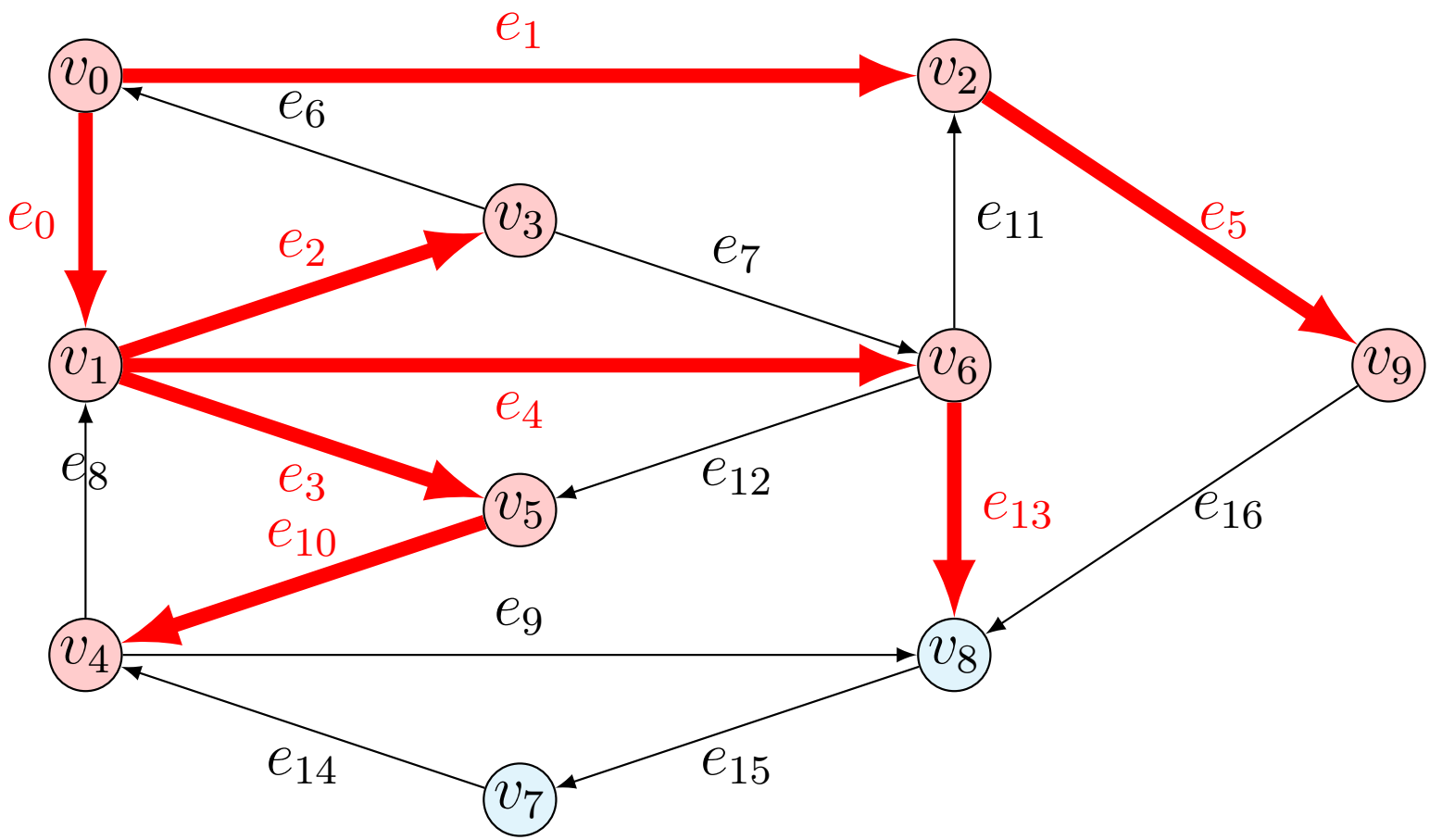
	現在の頂点	L	Q
0		$\emptyset$	$[v_0]$
1	v_0	$[v_0]$	$[v_1, v_2, v_3]$
2	v_1	$[v_0, v_1]$	$[v_2, v_3, v_5, v_6]$
3	v_2	$[v_0, v_1, v_2]$	$[v_3, v_5, v_6]$
4	v_3	$[v_0, v_1, v_2, v_3]$	$[v_5, v_6]$
5	v_5	$[v_0, v_1, v_2, v_3, v_5]$	$[v_6, v_4]$
6	v_6	$[v_0, v_1, v_2, v_3, v_5, v_6]$	$[v_4, v_8]$
7	v_4	$[v_0,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

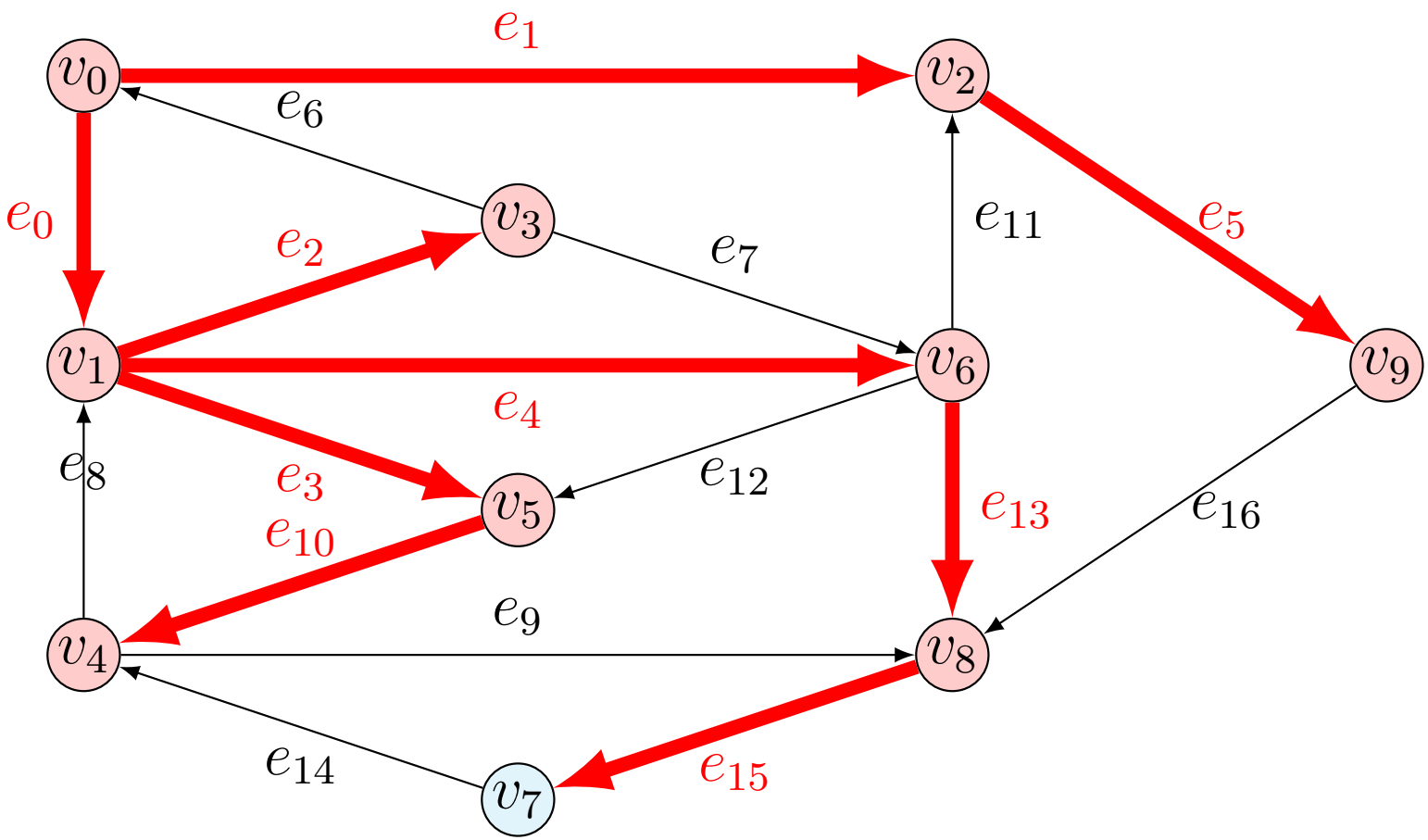
例 1.3: 結果

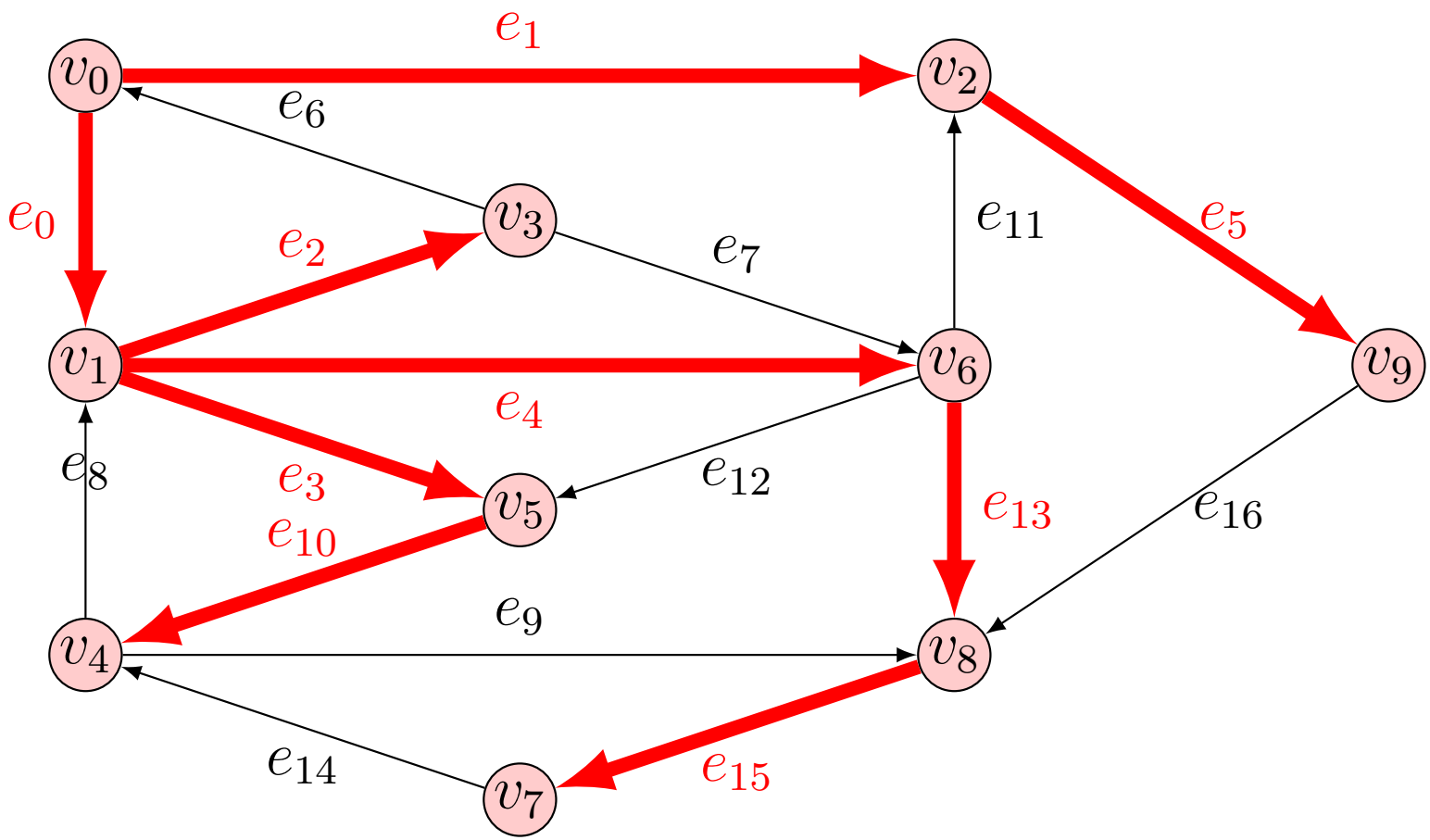

e_1
v_0
v_2
e_6
e_0
v_3
e_2
e_7
e_{11}
e_5
v_1
v_6
v_9
e_4
e_8
e_3
e_{12}
v_5
e_{13}
e_{16}
e_{10}
e_9
v_4
v_8
e_{14}
e_{15}
v_7


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
e_0
e_1
e_2
e_3
e_4
e_5
e_6
e_7
e_8
e_9
e_{10}
e_{11}
e_{12}
e_{13}
e_{14}
e_{15}
e_{16}

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	$[v_0, v_1, v_2, v_3, v_5, v_6, v_9, v_4, v_8, v_7]$	$\emptyset$