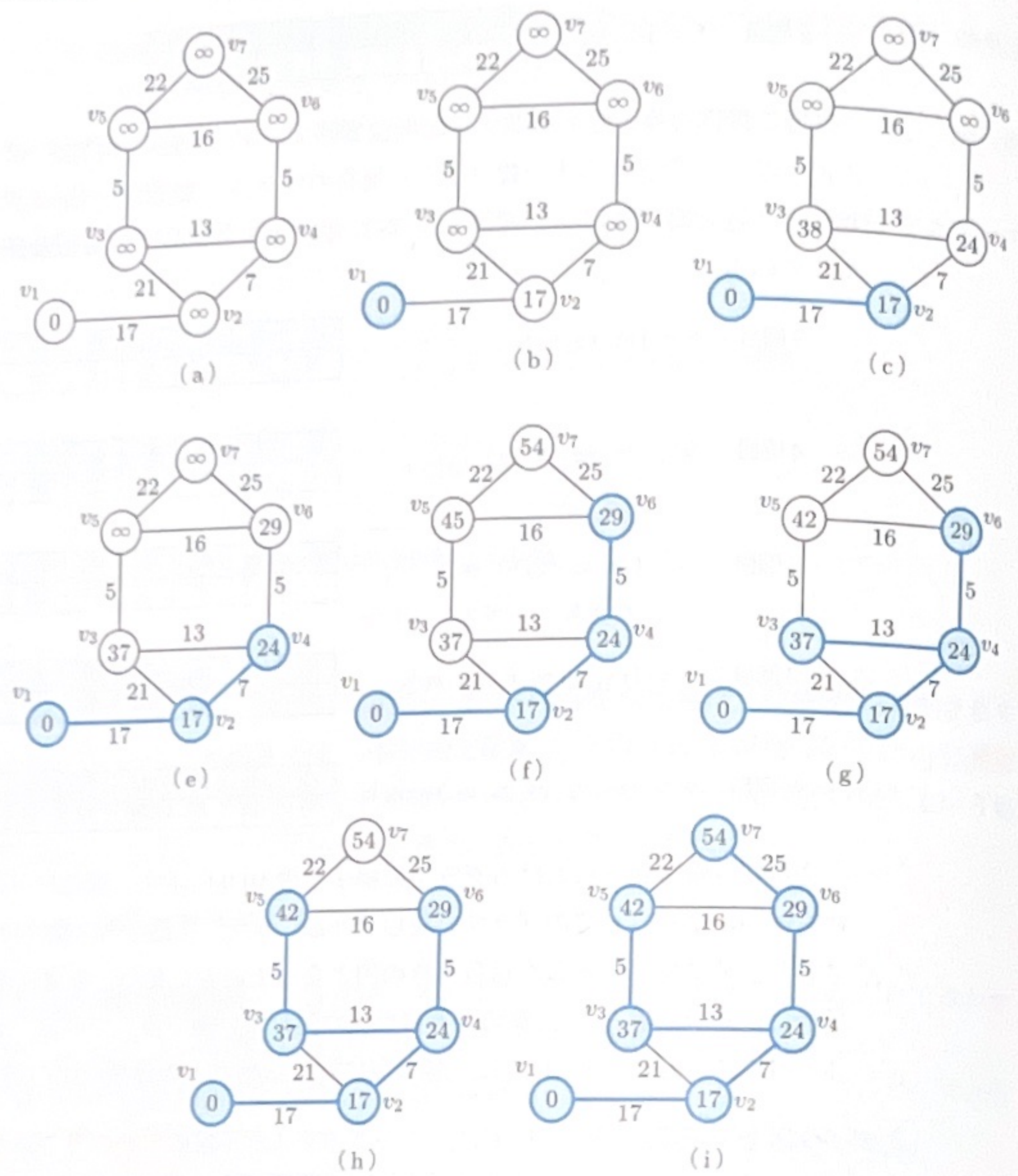

図 10.9 ダイクストラ法の実行例

算量を厳密に考えることは困難であるが，アルゴリズム中に二重の for 文が存在するので，時間計算量は少なくとも $O(n^2)$ であることがわかる．また，外側の for 文の中で実行される，"S に含まれない頂点の中から，配列 D の値が最小の頂点 v[k] を求める；" という行も，単純な方法ではすべての頂点について集合 S と配列 D をチェックする必要があるので，$O(n)$ の時間計算量が必要である．これらのことからわかるとおり，アルゴリズム全体の時間計算量は $O(n^2)$ であることが知られている．

また，第 5 章で説明したヒープを用いて集合 S に含まれる頂点の管理を行ったり，内側の for 文を削除して連結リストを用いて辺のチェックを行うなどの修正を加えることにより，アルゴリズムの時間計算量を $O(m \log n)$ にできることが知られている．ただし，辺の数 m の最大値は本章の最初で述べたように $\dfrac{n(n-1)}{2}$ なので，辺の数が多い場合はアルゴリズムの時間計算量は $O(m \log n) = O(n^2 \log n)$ となり，この修正

されたアルゴリズムは辺の数が少ないときのみ時間計算量が改善されるアルゴリズムであることがわかる．

なお，このダイクストラ法の説明では，簡単に理解できるように最短経路の長さのみを求め，実際の最短経路を求める部分については省略している．しかし，図 10.9 からもわかるように，このアルゴリズムを少し修正するだけで，簡単に最短経路自体も求めることができる．また，このダイクストラ法はグリーディ法のアイデアを使っているが，どのようなグラフについても正確に最短経路が求められることが理論的に証明されている．

第 10 章のポイント

1. **グラフ**とは，データの関係を視覚的に表す抽象概念であり，**頂点**および**辺**の集合で構成される．また，グラフに含まれる 2 つの頂点は，一般に頂点の列により結ばれており，この頂点の列のことを頂点間の**経路**とよぶ．
2. アルゴリズム中でグラフを格納する方法としては，隣接行列と隣接リストという 2 つのデータ構造が一般的に使われている．**隣接行列**は 2 次元配列を用いて各辺の情報を格納するデータ構造であり，辺が多いグラフを格納する場合に向いている．**隣接リスト**は，各頂点に対して，その頂点に隣接する頂点の情報を連結リストを用いて格納するデータ構造であり，辺の数が少ないグラフを格納する場合に向いている．
3. **グラフの探索**とは，始点となる頂点を指定し，始点からすべての頂点を 1 回ずつ調査する操作である．このグラフの探索については，**幅優先探索**と**深さ優先探索**という 2 つの方法が知られており，頂点数が n，辺の数が m の場合，どちらの探索方法も，グラフが隣接行列で表される場合は $O(n^2)$，隣接リストで表される場合は $O(n+m)$ という時間計算量で実行できる．
4. **最短経路問題**とは，与えられたグラフとグラフ中の 2 つの頂点間の最短の経路を求める問題である．この最短経路問題に対しては，**ダイクストラ法**という効率の良いアルゴリズムが知られており，このアルゴリズムを用いることにより，n 頂点のグラフに対して $O(n^2)$ 時間で最短経路を求めることができる．

演習問題

10.1 以下の文章の①～⑤について，それぞれ正しい記号を下から選べ．正しい記号が複数存在する場合はすべて列挙せよ．

情報科学の分野で用いられるグラフは（ ① ）であり，日常では（ ② ）などとして用いられている．また，このグラフは，（ ③ ）．

グラフを格納するための代表的なデータ構造としては，隣接行列と隣接リストがあるが，辺が少ないグラフを格納する場合は，記憶領域の面では（ ④ ）である．また，2 つの頂点間に辺があるかないかを頻繁に調べる場合は，（ ⑤ ）である．

①：a. $y = x^2$ などの関数の値を 2 次元平面上にプロットしたもの
　b. データの関係を視覚的に表す抽象概念
　c. 全順序関係をもつデータを特定の順序で記憶するためのデータ構造
　d. $O(1)$ 時間で探索を実行するためのデータ構造

②：a. 電車の時刻表　b. データ分布を表す円グラフ
　c. 高速道路の路線図　d. ネットワークの配線図

③：a. 頂点の集合と 2 つの頂点を結ぶ辺の集合で構成される
　b. 2 頂点間に必ず辺が存在する　c. 必ず根とよばれる頂点がある
　d. 辺の数が m の場合，頂点数は $m(m-1)/2$ 以下である

④：a. 隣接行列のほうが有利　b. 隣接リストのほうが有利
　c. 隣接行列でも隣接リストでも同じ　d. 隣接行列や隣接リストでは不十分

⑤：a. 隣接行列のほうが有利　b. 隣接リストのほうが有利
　c. 隣接行列でも隣接リストでも同じ　d. 隣接行列や隣接リストでは不十分

10.2　深さ優先探索のアルゴリズムを，再帰アルゴリズムとして記述せよ．

10.3　図 10.2 (a)のグラフに対して，始点を v_3 として幅優先探索と，深さ優先探索を実行せよ．なお，図 10.5 (d)，および，図 10.6 (d)同様の図を示すこと．

10.4　図 10.10 のグラフに対する以下の問いに答えよ．

(1)　このグラフを表す隣接行列を示せ．

(2)　このグラフを表す隣接リストを示せ．

(3)　このグラフに対してダイクストラ法を適用し，頂点 v_1 からその他の頂点への最短経路の長さを求めよ．

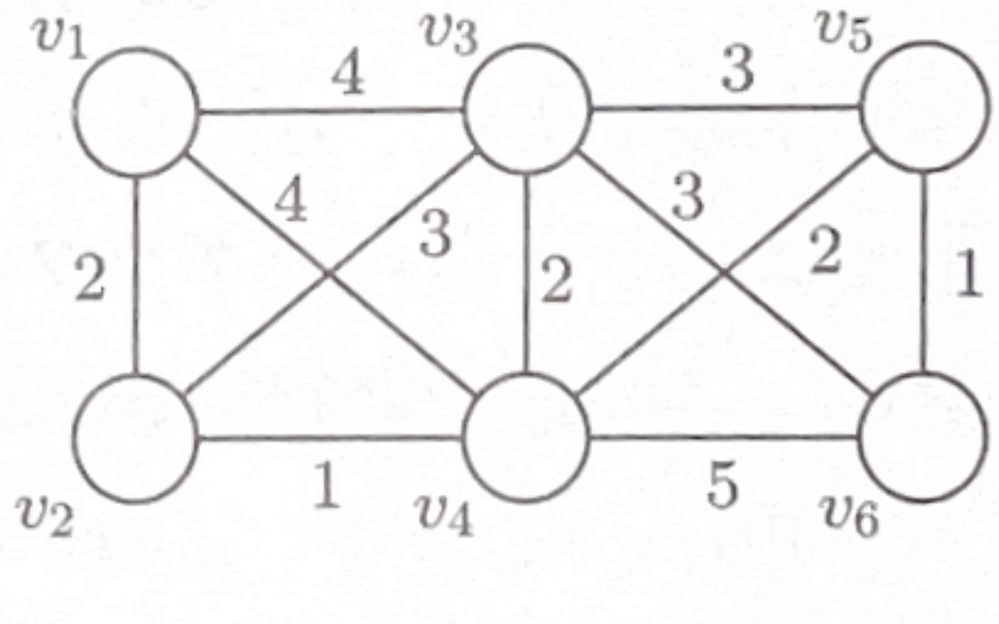

図 10.10

10.5　ダイクストラ法は辺の重みが負である場合は正常に動作しない場合がある．ダイクストラ法が正常に動作しないような，負の重みをもつ辺が存在するグラフの例を示せ．