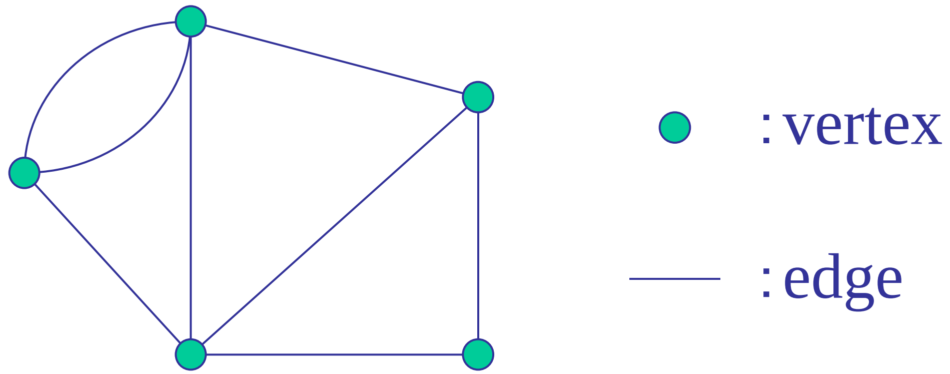
Dijkstra法によるグラフ最短路問題

グラフとは

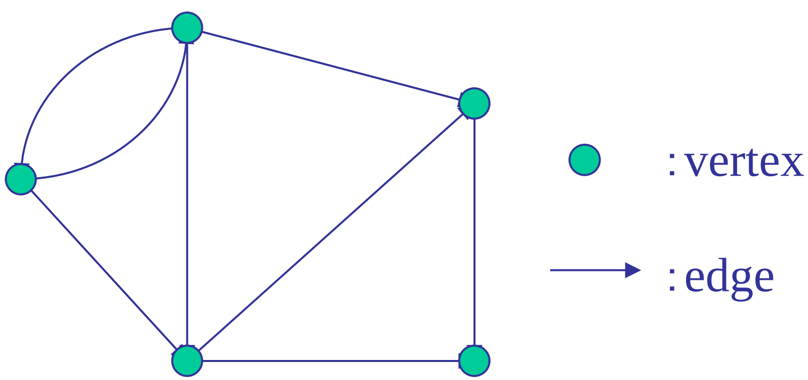
・グラフG(Graph)は、節点の集合V(Vertex)と、それを結ぶ枝E(Edge)の集合から成る（下図）

・G=(V,E)で表される

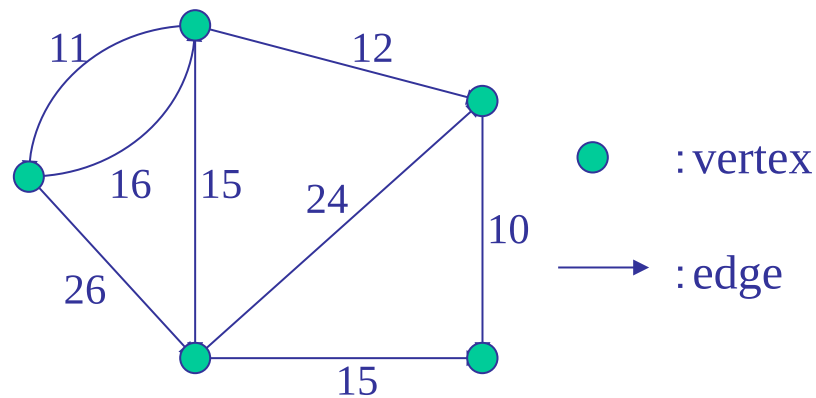


グラフ

グラフには、それぞれの枝に向きのある有向グラフ(directed graph)、向きのない無向グラフ(undirected graph)、枝に重みwのついたネットワーク(Network)N=(G,w)などがある．重みwというのは、節点間の距離みたいなもの．



有効グラフ



ネットワーク

これらのグラフは、アルゴリズム解析の道具やデータ構造として用いられる．例えば、グラフを用いた問題として一筆書き出来るかの問題や，最短経路探索の問題などがある．

最短経路問題

・最短経路問題は、グラフ問題の一つ

・重みつきグラフ G=(V,E) が与えられた時に、任意の2頂点を結ぶ経路の中から、辺の重みの総和が最小のものを求めるもの

・ カーナビで今の場所から目的地への最短の道順の探索などに利用

最短経路問題のクラス

- 最短経路問題は、解を求める範囲によって三つのクラスに分けることができる

１. 出発点Aから目的地Bへの最短経路を求める

２. 出発点AからV の中の各頂点への最短経路のコストを全て求める

代表的なアルゴリズム：Dijkstraのアルゴリズム

３. V の中の全ての頂点対の最短経路のコストを求める

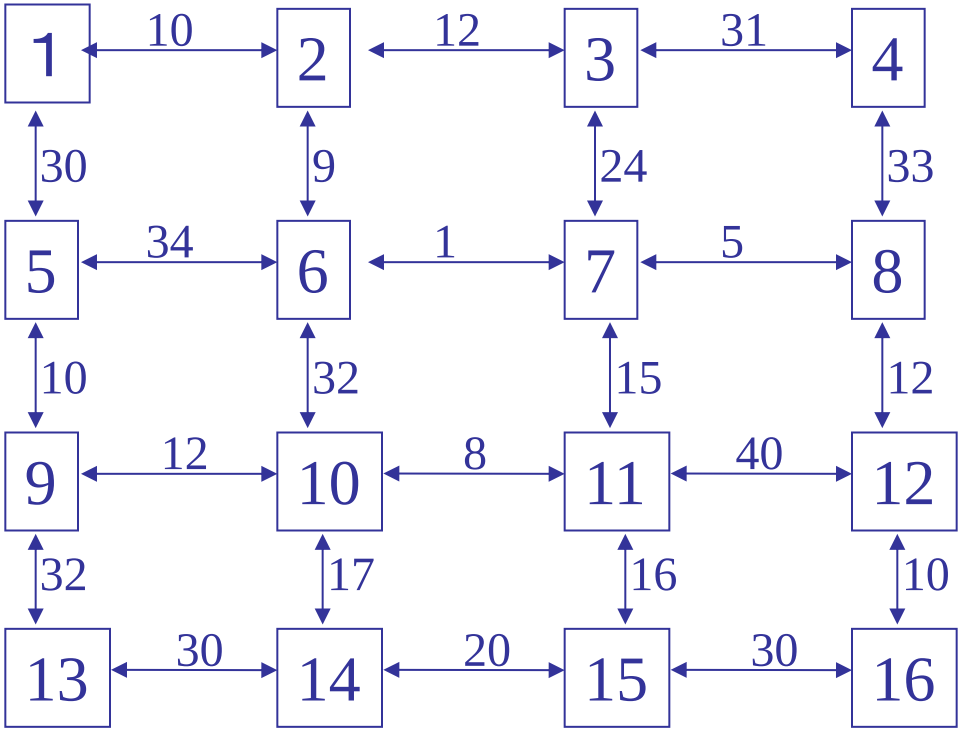
代表的なアルゴリズム：Floydのアルゴリズム

Dijkstra法

- 1節点から全節点への最短経路を求めるアルゴリズム
- 1959年にE.Dijkstraによって提案された
- 全ての枝の重みが非負の場合にのみ適用できる
- 手順:
  １．始点のラベルを0、それ以外の点のラベルを無限大とする
  ２．最短路の長さが確定した点のラベルを 確定ラベル、それ以外の点を一時ラベルと置き、次に一時ラベルの中で最小の点xをみつける
  ３．２で見つけた点を確定ラベルに変更し、隣接する点の一時ラベルを、「一時ラベルの値と、点xのラベルの値とその間の枝の重みを足したものの小さいほう」の値に変更し、確定ラベルにする
  ４．１，２，３を繰り返し、すべての点が確定ラベルになったら終了．その結果，各点の確定ラベルが始点からの最短路の長さになっている。

Dijkstra法による最短経路検索

下図の有向グラフにおいて, [13]から[4]までの最短経路を、Dijkstra法を用いて求めてみる



まず、このグラフについてのデータを与える

(始点・・・Ns=13,終点・・・Ng=4)

存在する区間データを与える

区間データ・・・(Ni,Nj,2点間の距離)

1 2 10

2 1 10

2 3 12

3 2 12

:

15 16 30

以下の3つのリストを用意する

- リストA (未調査リスト)
- リストB (調査中リスト)
- リストC (調査済リスト)

1. 区間データからノードデータ(Ni,Nsからの最短距離)を生　成しリストAに登録、最短距離の初期値は∞

リストA:(1,∞)(2,∞)(3,∞)・・・(16,∞)

2.始点Nsに関するノードデータを未調査リストから調査済リストへ移動、その際ノードデータの最短距離を0に更新

リストA:(1,∞)(2,∞)(3,∞)･･･(16,∞)

リストB

リストC(13,0)

3.区間データを元に、始点Nsから直接到達可能なノードを調べ、そのノードに関するノードデータをリストAからリストBに移動しノードデータを更新

リストA:(1,∞)(2,∞)(3,∞)・・・(16,∞)

リストB(9,32)(14,30)

リストC(13,0)

4.以下の項目をリストBが空になるまで繰り返す。

(a)リストBから最短距離最小のノードNmを選びCへ移動

リストA:(1,∞)(2,∞)(3,∞)・・・(16,∞)

リストB(9,32)

リストC(13,0)(14,30)

(b)Nmから直接到達可能なノードNiを探し、NiがリストA にあればリストBへ移動

リストA:(1,∞)(2,∞)(3,∞)・・・(16,∞)

リストB(9,32)(10,∞)(15,∞)

リストC(13,0)(14,30)

(c)NiがBにあれば、NsからNmの最短距離に区間距離Nm,Niを加えた値と、既知の最短距離Ns,Niを比べ、より小さい方を新たな最短距離とする

リストA:(1,∞)(2,∞)(3,∞)・・・(16,∞)

リストB(9,32)(10,30+17)(15,30+20)

リストC(13,0)(14,30)

5.リストBが空に成った時点でA,B,Cについて以下のようなリストが得られる。

リストA:

リストB:

リストC:
(4,120)(8,102)(12,90)(3,89)(16,80)(2,77)(1,72)(6,68)(7,67)(11,52)(15,50)(10,44)(5,42)(9,32)(13,0)(14,30)

ここでリストCに、始点Nsから各ノードへの最短距離が入っていることになり、ノード13から4までの最短距離が120であることが判明する

ノードデータに対して一歩手前のノード番号を記憶させておき、終点から順に遡ることで最短距離だけでなく，最短経路も求めることができる