2-2　データ表現

東京大学 数理・情報教育研究センター

2021年4月6日

概要

- コンピュータでデータを扱うためのデータ表現の基礎を学びます。

本教材の目次

１．構造化データ　4
２．非構造化データ　5
３．データの表現方法　数の場合　6
３．データの表現方法　文章（文字）の場合　8
３．データの表現方法　音声の場合　11
３．データの表現方法　画像・動画の場合　17
３．データ構造（配列、木構造、グラフ）　21

構造化データ

- コンピュータで扱うデータと聞いてどの様なものを想像しますか？実はデータには様々な形式のものが存在します。
  - 例えば、何らかの製品を販売している会社が扱っているデータを考えてみましょう。この場合、社内の各部署のコストをまとめたデータや、いつどこにどれだけ製品を出荷したのかを記録したデータが考えられます。部署や「どこ」に通し番号を振っておけば、これらは数値で表すことができるデータと考えられます。
- この様な数値によるデータは表（ひょう）形式で管理することができ、構造化データと呼ばれます。

非構造化データ

- 一方、近年、ネットワーク技術の進歩、ネットワーク上のサービス（Twitterなどのソーシャルサービス、メールやストレージのクラウドサービスなど）の発展、携帯端末の普及などにより様々なデータが大量に作成され、扱われる様になりました。
- 例えば、新聞・Webなどの文章データ、写真などの画像データ、電話などの音声データ、テレビなどの動画データなどを挙げることができます。こういった単純には数値では表すことができない様なデータを非構造化データと言います。
  - 例えば、何らかの製品を販売している会社では、業務日誌・議事録などの文章データ、通話記録などの音声データ、TV会議などの画像データや動画データなどの非構造化データが発生することが考えられます。

データの表現方法　数の場合

- コンピュータ上でデータを表現するにはどうしたら良いでしょうか？　この教材にも数値や文章など様々なデータ（値）が含まれていますが、これらの値をコンピュータは2を基数とする数、すなわち、0と1の2つの数から構成される2進数を用いて表しています。
- 我々が普段使っている数は0から9の数を用いる10進数です。10進数では10を基準とする数（基数と言います）で数が表現されています。
- 例えば、794という数は、各桁毎に右（1の位）から順に見ていくと、1（10の0乗）が4個、10（10の1乗）が9個、100（10の2乗）が7個からなる数として構成されていると見なせます。

2進数

- 2進数は2を基数としています。つまり、2の0乗が●個、2の1乗が◎個、2の2乗が△個、…という様な形で数を表すことになります。
  - 例えば、2進数で1101と記述すると、10進数と同様に各桁毎に右から順に、1（2の0乗）が1個、2（2の1乗）が0個、4（2の2乗）が1個、8（2の3乗）が1個からなる数として構成されています。つまり、2進数の1101は、10進数で表すと
    $1\times1+2\times0+4\times1+8\times1=13$となります。
- 10進数の1101と2進数の1101がこのスライドの様に一緒に書かれていると非常に紛らわしいですね。そこでこれらを区別するために、基数を添え字として明記する記法が存在します。例えば、10進数の場合は$1101_{(10)}$、2進数の場合は$1101_{(2)}$となります。この記法を用いた場合、$1101_{(2)}$は10進数では$13_{(10)}$と表すということです。

データの表現方法　文章（文字）の場合

- ここまでの話で、コンピュータは2進数を使っており、我々が普段用いている10進数の数も2進数で表すことが出来ることが分かりました。
- では、数字ではなく文章、つまり文字はコンピュータはどの様に表すのでしょうか？　実はやはり2進数を用いて表しており、数字と文字を一対一で対応付けて考えています。
  - 例えば、Aは$0_{(10)}$、Bは$1_{(10)}$、Cは$2_{(10)}$、…、Zは$25_{(10)}$と対応付ければ数で文字を表すことが出来ることが分かりますね。この対応付けでは、アルファベットのMは$13_{(10)}$ですので、コンピュータ上ではMを表すのに$1101_{(2)}$という数を用いれば良いことになります。

文字コード

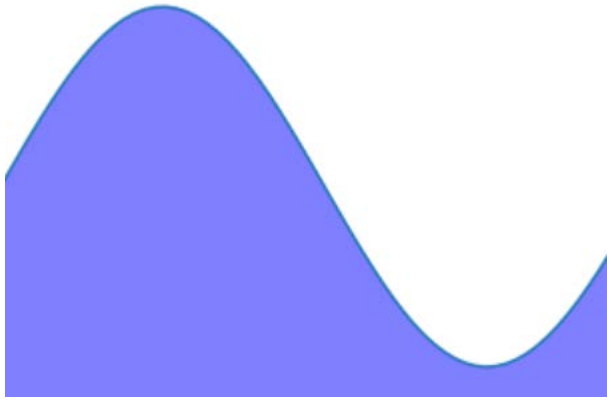
- 実際にコンピュータが用いている数字と文字の対応付けを文字コードと言いますが、7桁の2進数でアルファベットや記号などを表すASCIIと呼ばれる文字コードがあります。
  - ASCIIでは、例えば$65_{(10)}$とA、$90_{(10)}$とZ、$97_{(10)}$とa、$122_{(10)}$とzが対応付けられており、記号では、$33_{(10)}$と!、$63_{(10)}$と?、$61_{(10)}$と＝などの様に対応付けられています。
  - 余程のことがない限り、どの文字にどの様な数が対応付けられているかは覚える必要ありません。
- ASCIIの対応付けには日本語は含まれていません。日本語を含む文字コードとしては、UTF-8、JIS、 Shift_JIS（シフトJIS）などを挙げることができます。

情報の単位と量

- 2進数も10進数と同様に数字ですので、「$1101_{(2)}$は4桁の値」と表現できます。ですが、コンピュータ上で表す場合には特にこの1桁分をビット (bit) という単位で数えます。つまり、「$1101_{(2)}$は4ビットの値」であると言えます。先程の文字コードはASCIIは7桁でしたので、7ビットで文字や記号を表していることになります。
- また8ビットを1バイト (Byte, Bと略記) と言います。
  - スマホなどの携帯端末のデータ通信の量を表すのにメガやギガという単位が出て来ますが、1キロバイト (KB) ＝1024バイト (B) 、1メガバイト (MB) ＝ 1024KB、1ギガバイト (GB) ＝1024MB、1テラバイト (TB) ＝1024GB、…という具合に大きくなっていきます。
- コンピュータ上でデータをやり取りする際に、そのデータにどれだけの情報（情報量）が含まれるかについて議論する場合、ビットやバイトという単位を用いて議論するのが一般的です。

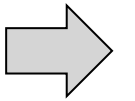
データの表現方法　音声の場合

- では、音（声）をコンピュータで扱う場合、どの様に表しているのかと言うと、やはり2進数を使って数値で表しています。しかし、例えばスピーカの音量の大小を数値で表すというのは分かりますが、音を数値で表すと言っても良く分からないと思います。
- 実は、音は空気の密度が濃くなったり薄くなったりする波によって我々の耳に届いています。つまり、この波（疎密波と言います）がどの様な形をしているのかを数値で表してコンピュータで扱う様にすれば良いのです。



疎密波

どう数値化する？



http://iconhoihoi.oops.jp/item/category/pc/

アナログとデジタル

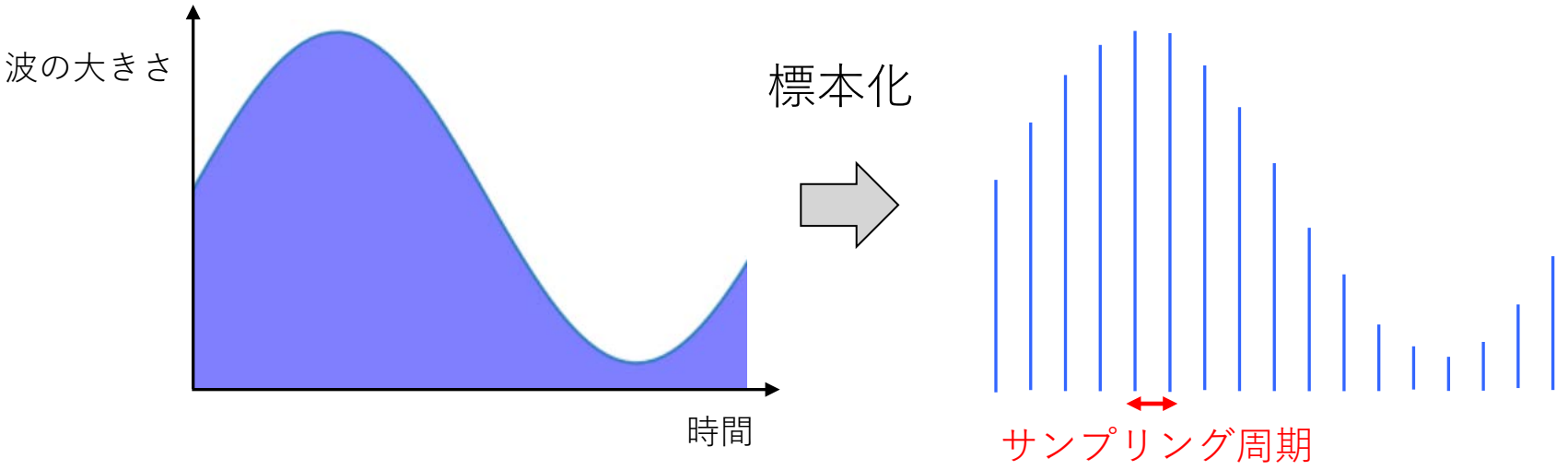
- この様な波は連続的な値で構成されています。一方でコンピュータが扱う値は離散的である必要があります。前者をアナログの値と呼び、後者をデジタルの値と呼びます。
  - 例えば、体温計には、水銀の伸びと体温計本体に刻まれた目盛りから体温を目測する水銀体温計と「36.5°C」の様に一定の桁数で計測した体温を表示してくれるデジタル体温計（電子体温計）がありますね。後者は名前にもデジタルと入っていますが、（例えば）小数点1桁までの離散的な値で結果を表示するデジタルの値になっています。一方前者は水銀が伸びたり縮んだりするものの、水銀が0.1°C刻みで結果を表示する訳ではなく、必ず連続的な値で結果を表示します。
  - 同じ人がそれぞれの体温計で体温を測った場合、水銀体温計は「36.524…°C」の様な結果となるのに対し、デジタル温度計は「36.5°C」の様な結果を表示します。

標本化と量子化

- つまり、コンピュータが音声（つまり、波）を扱うには、アナログの値からデジタルの値に変換してやる必要があります。その為には波に対して標本化と量子化と呼ばれる操作を行います。この2つの操作をまとめて音声の符号化と呼びます。
  - 大まかに言うと、波を横方向に離散化することを標本化と言い、縦方向に離散化することを量子化と言います。

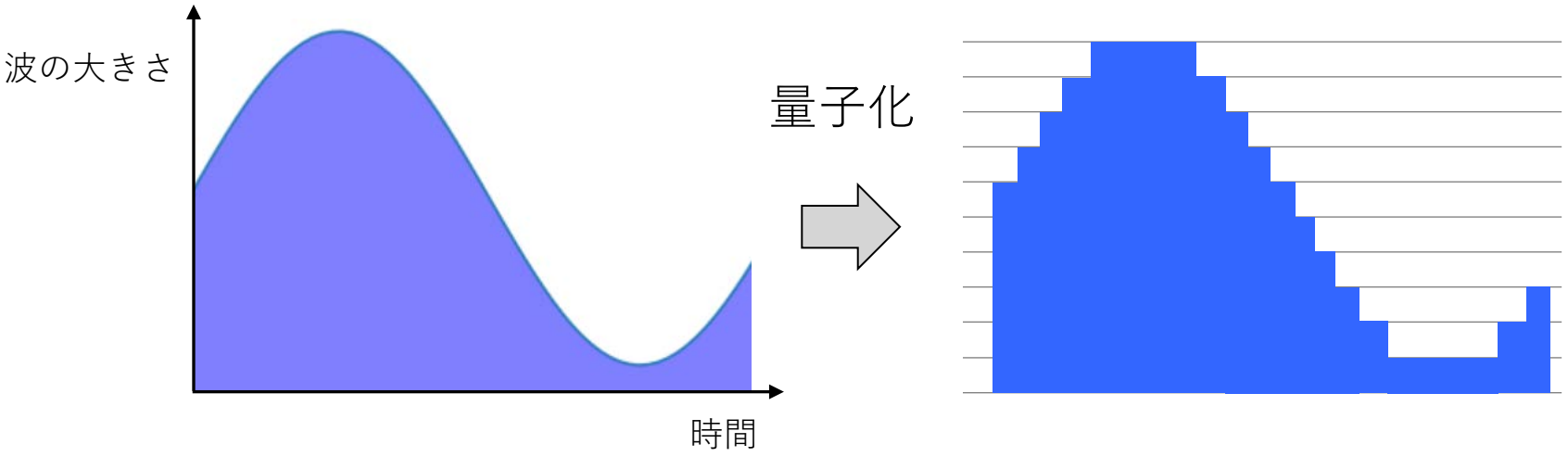
標本化

- 標本化は波の値を取得（記録）する間隔を決める操作です。例えば、30秒間にわたって発生している音（波）は、元のアナログの値ではその30秒間のどの時間（ 11秒目だろうと25.78…秒目だろうと）を見ても波の値を取得することができます。標本化では、例えば5秒間隔（すなわち、0秒目、5秒目、10秒目、…、25秒目、30秒目）**のみ**波の値を取得してコンピュータに与える（記録させる）ことにします。
- 1秒間に何回値を取得したかを表す値をサンプリング周波数と言います。サンプリング周波数の逆数をサンプリング周期と言います。



量子化

- 量子化は波の大きさ（量）の値を離散的な値に変換する操作です。例えば、元の波の各値の大きさは0以上100以下の値であった場合、波の大きさは24.5であったり、82.334…であったりするのですが、量子化ではそれを例えば0、10、20、…、90、100の11通りの値に制限してしまう操作になります（つまり、24.5→20、82.334…→80の様に変換してしまいます）。



PCM

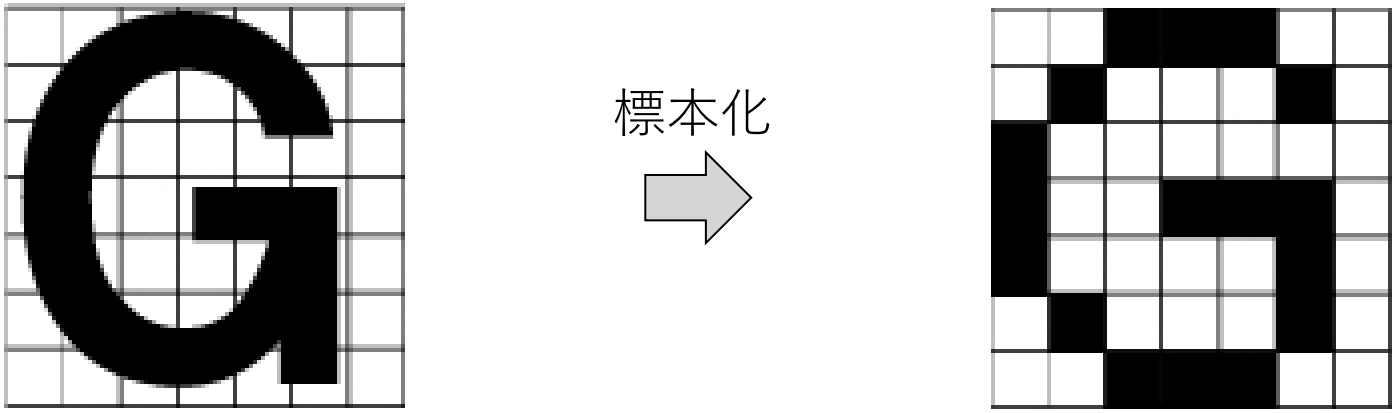
- 標本化の際の値を取得する間隔は、元の波を上手くコンピュータ上で再現可能な様な間隔にする必要があります。具体的には元の波の周波数（1秒間に繰り返される波の数）の2倍以上になる様にサンプリング周波数を設定すれば良いことが知られています。
- 代表的なデジタルへの変換方式にPCM（Pulse Code Modulation, パルス符号変調）があります。PCMはCD, DVD, mp3ファイルなど多くのメディアやファイルで利用されています。
  - 例えば、一般的なCDはサンプリング周波数を44.1kHzとし、値を2^{16}=65536段階、すなわち、16ビットの値に変換して記録されています。

データの表現方法　画像・動画の場合

- ここまでの話の流れで想像がつくと思いますが、画像や動画の場合もコンピュータは2進数を用いて数値で表しています。どの様に数値で表すのかと言うと、音声の場合と同様に、標本化と量子化を行います。やはり、この2つの操作をまとめて画像の符号化と呼びます。
- 動画はパラパラ漫画の様に画像を大量に使えば実現できると考えて、この教材では画像の場合のみ説明します。

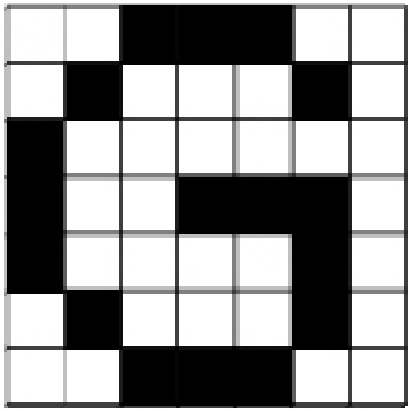
標本化　画像の場合

- 音声の場合は波があり、標本化ではその波の値を取得する間隔を決めていました。つまり、波を一定間隔で区切っていました。これと同じ様に、画像でも画像を縦と横に一定間隔で区切って値を取得する位置を決めて標本化を行います。
- この区切った四角形の部分を画素（ピクセル）と言います。画素1つにつき1つの色を表す様にします。その為、画素の数（解像度と言います）が多い程、元の画像の情報を保存できるわけです。
  - 例えば以下の例では、アルファベットの「G」を7×7個の画素で構成する様に標本化しています。

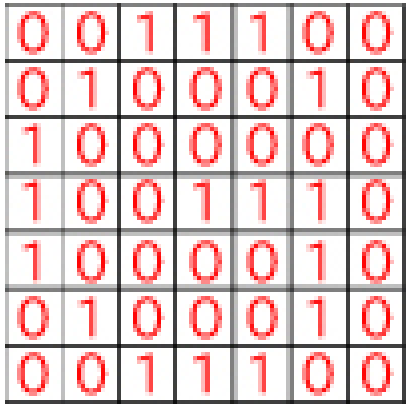


量子化　画像の場合

- 音声の場合、量子化するのは波の大きさでした。画像の場合は、画素を構成する数値は色になります。
- 簡単のため、まず白黒の画像を考えましょう。色は白と黒の2通りしかありませんので、画素をなるべく小さく取ってしまえば、画素は白か黒しかありません。そこで白い部分を0、黒い部分を1で表せば、画像の各画素を1ビットで量子化することが出来ます。
  - 例えば以下の例では、各画素をGの字が存在する画素を1、存在しない画素を0で表しています。

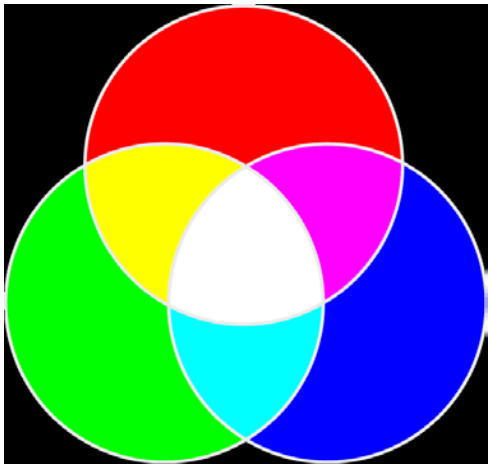


量子化



RGB

- 白黒画像ではない場合に様々な色を表す為の代表的な方法にRGB方式があります。Rは赤 (Red) 、Gは緑 (Green)、Bは青 (Blue) を表しており、コンピュータは光の三原色である赤・緑・青の3色を混ぜあわせてこれらの割合を調整することで様々な色を作成しています。
  - 例えば、赤：緑：青＝1：1：0の場合は黄色になり、赤：緑：青＝1：1：1の場合は白になります。全てを0にすると黒になります。
- 赤・緑・青の量をそれぞれ256段階で量子化した場合、1つの画素の色は8ビット×3=24ビットで表すことが出来ます。

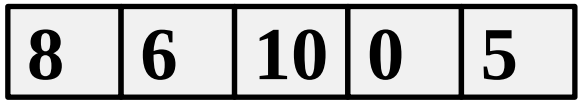


データ構造

- コンピュータが様々なデータを2進数の値として扱っていることは既に述べました。これらのデータをアルゴリズムで扱ったり、プログラムなどで処理したりする場合には、どの様にデータが格納されているのかを考える必要があります。このデータを格納する方法をデータ構造と言います。
- この教材では代表的なデータ構造について説明します。
  - 配列
  - 木構造（ツリー）
  - グラフ

配列

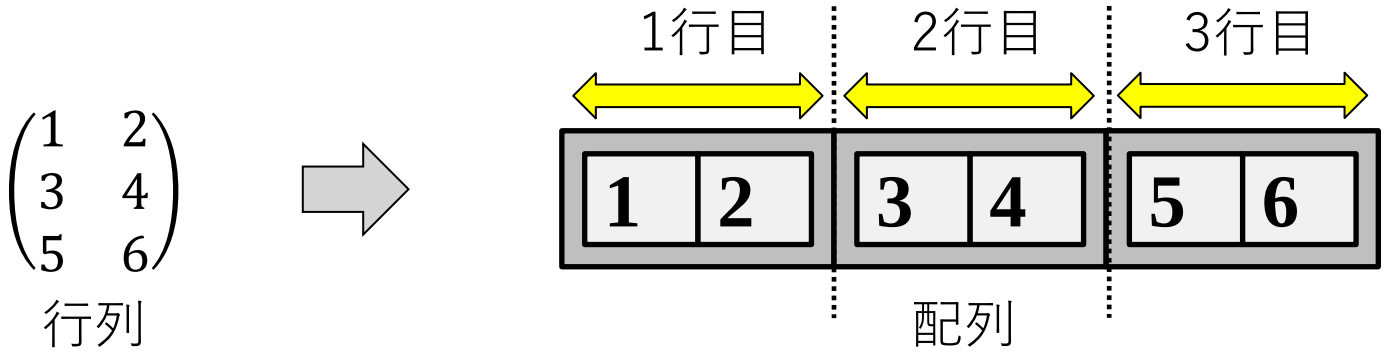
- 配列は値を順序付けて格納するデータ構造です。
- 「順序付けて格納する」という表現はやや複雑ですが、要するに配列に格納されているデータ（値）は「〇番目のデータ（値）」と呼ぶことが出来る形式で保管されているということです。
  - 配列は数学のベクトルを表すのに適したデータ構造と言えます。例えば、(1,2)と(2, 1)というベクトルは全く異なる値ですよね。このベクトルの様に値が順序付けられている場合、配列を用いると良いでしょう。
  - この教材では配列は下の様な図で配列を表すことにします。この例の配列では1番目の値は8で、4番目の値は0が格納されています。



- 配列は探索や並べ替えなど様々なアルゴリズムにおいて用いられます。

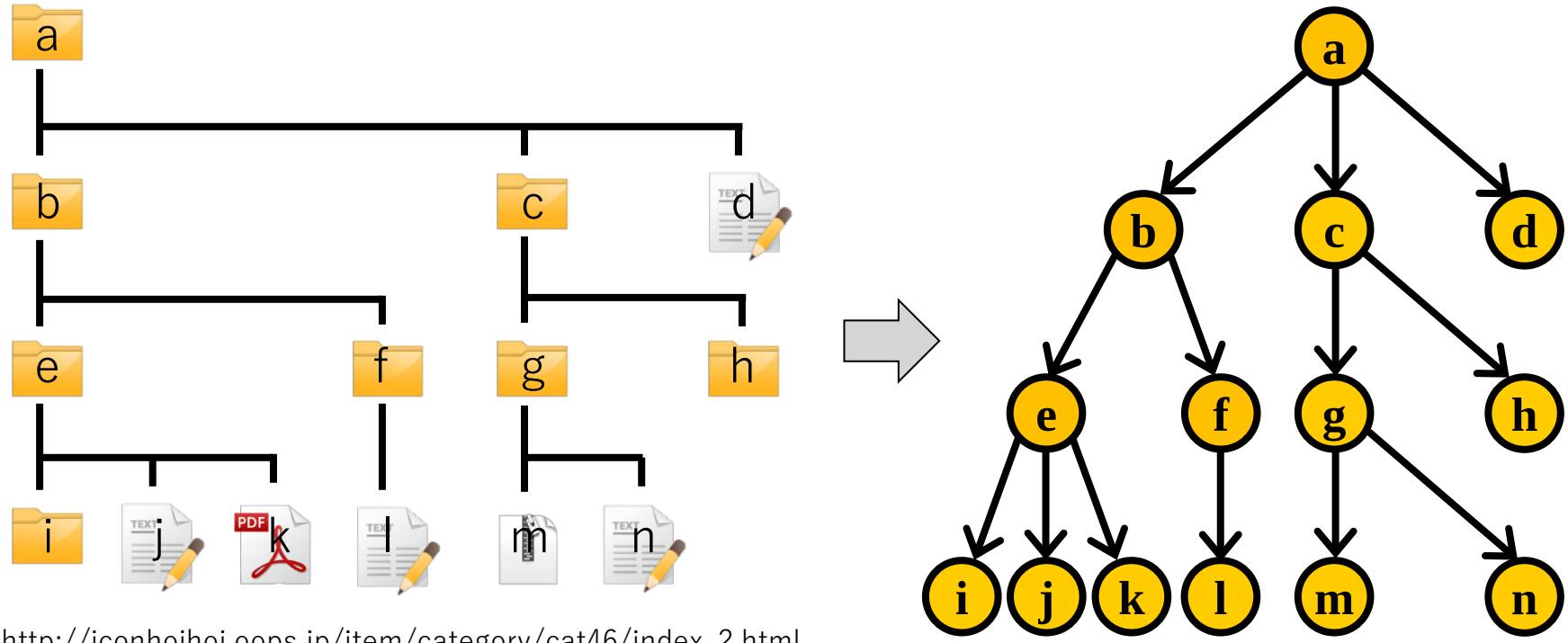
配列

- 配列には数値だけではなく、様々なデータを格納することができると考えるのが一般的です。その際、配列の中に配列を格納している場合、多重配列（多次元配列）と呼びます。
  - 数学の行列は多重配列を用いて表すことが出来ます。行列の各行の値を1つの配列に格納します。そして、それらの配列を1つの配列に順番に格納します。
  - 例えば、以下の例では3×2の行列に対して、1つの行毎に配列に格納し、更にその3つの配列を1つの配列に順に格納しています。



木構造

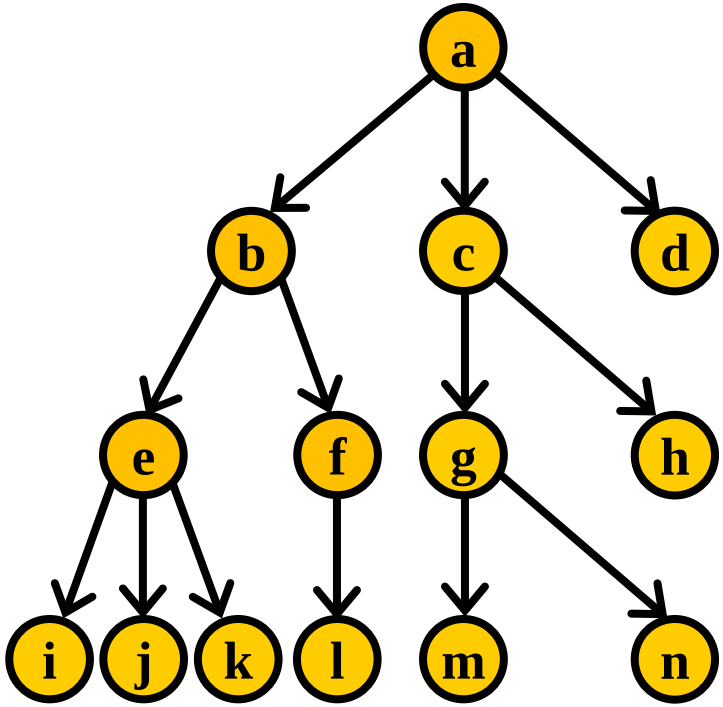
- 木構造（ツリー）は樹形図状にデータを格納するデータ構造です。組織図や家系図、コンピュータ上のファイル・フォルダの階層構造などは樹形図状に表現できますが、こういった様々な分野で用いるデータ構造です。
- 例えば、以下はファルダ類の階層構造の例ですが、この教材では右の図の様な丸と矢印からなる図で木構造で表すことにします。



http://iconhoihoi.oops.jp/item/category/cat46/index_2.html

木構造について

- 木構造において用いる簡単な用語を説明します。
- 図の丸 ● で表す部分を点と呼び、矢印を枝（もしくは辺）と呼びます。矢印が1本も入って来ない点を根と呼びます。下の例では点aが根です。ある点から出た矢印の先の点をその点の子と呼びます。下の例では点aの子は点b,c,dの3つです。

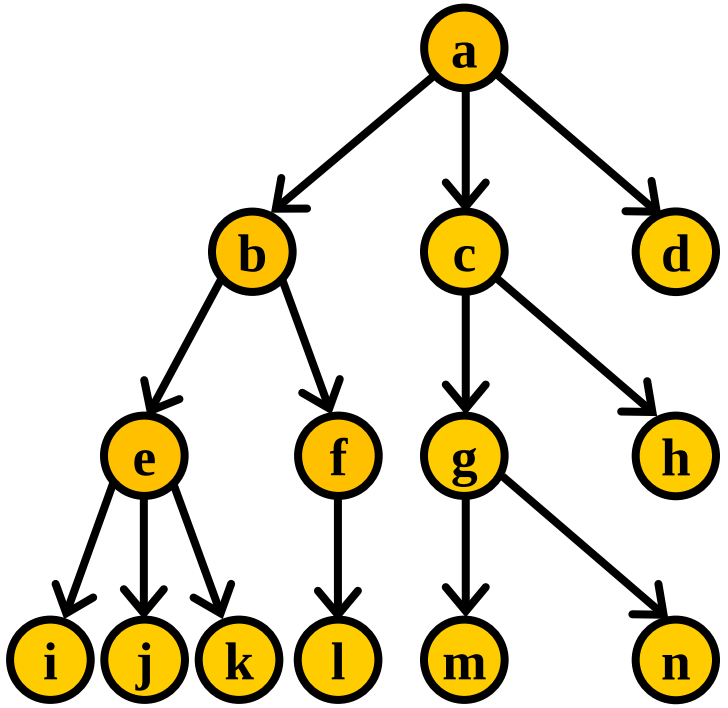


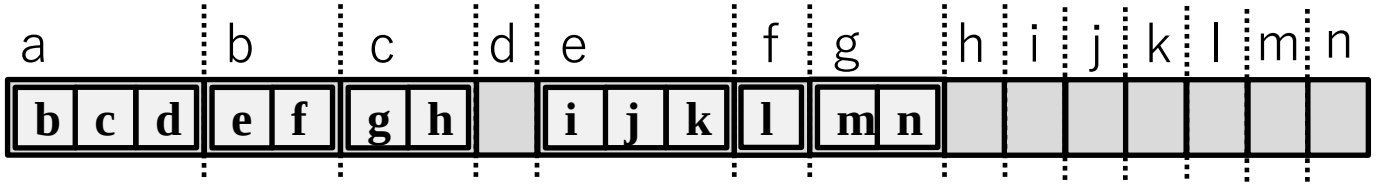
木構造の表現方法

- 木構造の代表的な表現方法は以下の2つです。いずれも多重配列を用いて実現することができます。
  - 隣接リスト
  - 隣接行列

隣接リスト

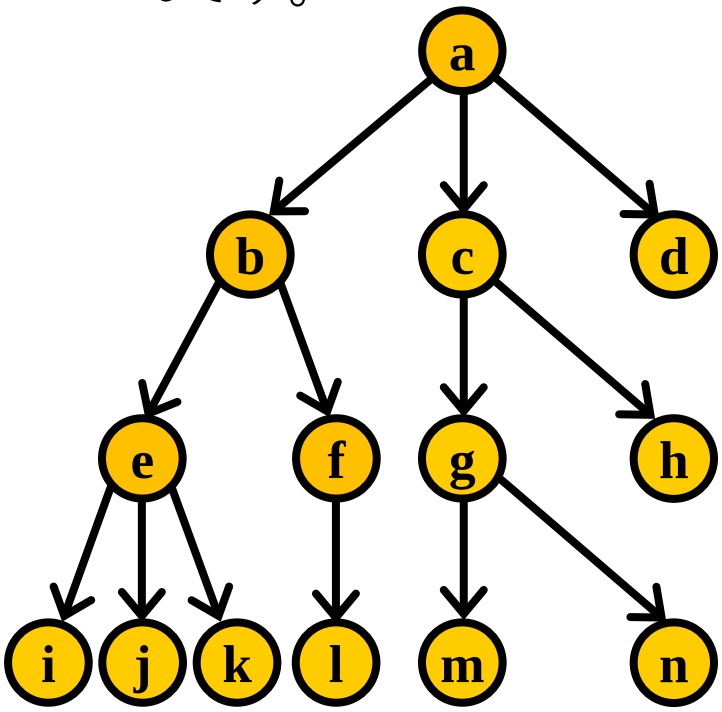
- 隣接リストは、各点毎に1つの配列を用意し、その点の子を全て格納します。格納する順番は自由ですが、利便性から何らかの順序（例えば、昇順・降順など）に従って格納することが多いです。
- 例えば、右の木構造は下の様な多重配列による隣接リストで表します。左端が点aの子（の名前）を格納した配列を格納しています。その後、b,c,d…と順に各点の子を格納しています。
- dは子がありませんが、それでも4番目は何も格納しないd用の領域になっています。





隣接行列

- 隣接行列は、その名前の通り行列を用いて木構造を表します。既に述べた通り、行列は多重配列で表すことができます。この行列ではx行y列の値が1だと、点xから点yへの枝がある、すなわち、点xの子がyであることを意味します。
- 例えば、aの行はb,c,d列だけが1であり、それ以外は全て0です。



	a	b	c	d	e	f	g	h	i	j	k	l	m	n
a	0	1	1	1	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	1	1	0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	1	1	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	0	0	0	0	1	1	1	0	0	0
f	0	0	0	0	0	0	0	0	0	0	0	1	0	0
g	0	0	0	0	0	0	0	0	0	0	0	0	1	1
h	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0

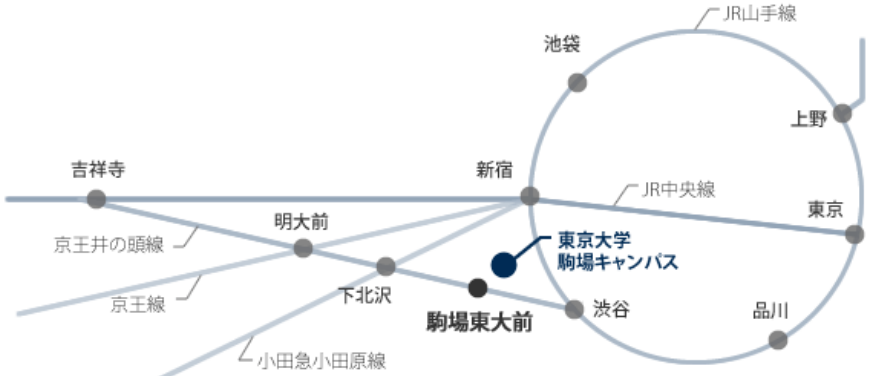
行列

グラフ

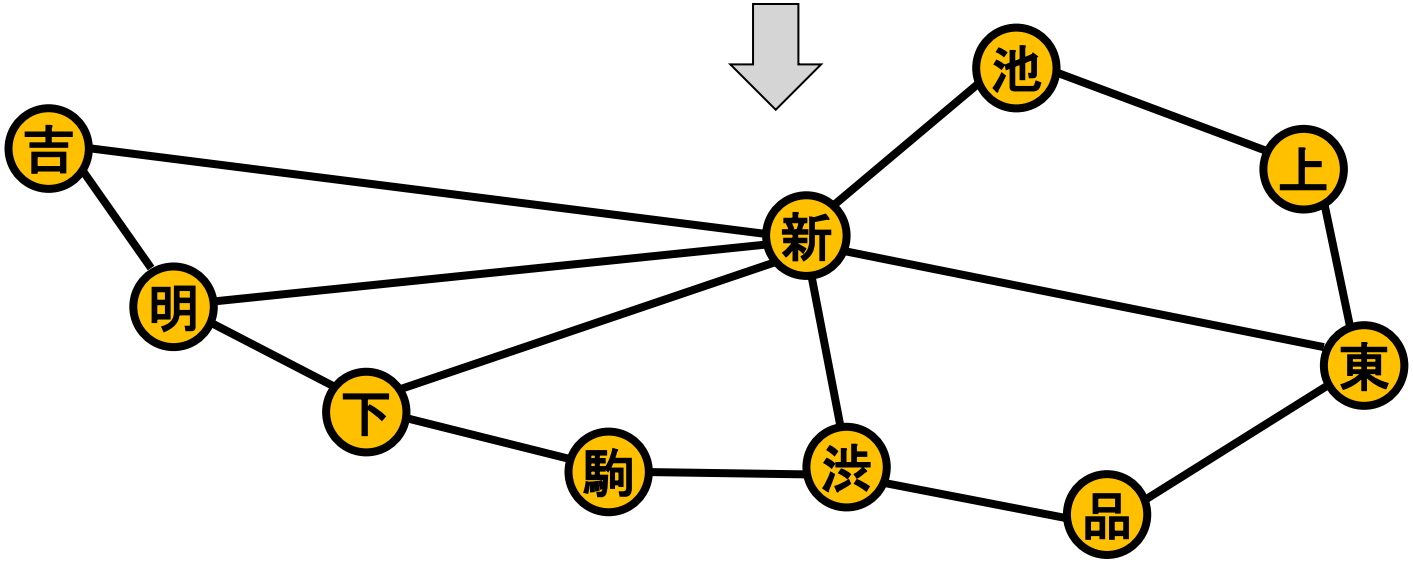
- グラフはネットワーク状にデータを格納するデータ構造です。交通網やSNSの接続関係や、別の章において出てくるニューラルネットワークなどもグラフで表すことが出来ます。
  - グラフというと折れ線グラフや棒グラフを想像する人もいると思いますが、そのグラフとは違います。

グラフ

- 例えば、下の図は東大周辺の電車の路線図ですが、駅を点、線路を枝と見なすことでグラフで表すことが出来ます。

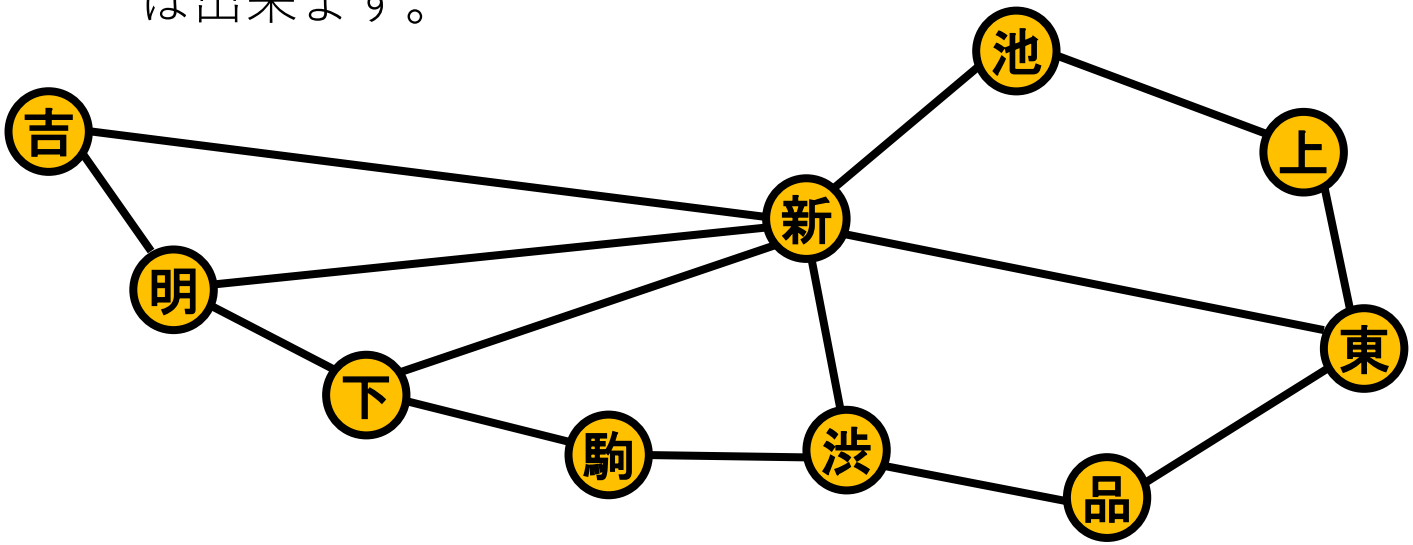


https://www.c.u-tokyo.ac.jp/info/about/visitors/maps-directions/index.html



グラフ

- このグラフの例が木構造と似ていることに気が付いたでしょうか？実はグラフは木構造を一般化したものです。
- 木構造と同様に、図の丸 ● で表す部分を点と呼び、矢印を枝（もしくは辺）と呼びます。
- 木構造は幾つかの異なる点をたどって同じ点に戻ってくることは出来ない様な構造になっていましたが、その様なことが可能である構造の場合、グラフと呼ぶことになります。
  - その様なことが出来ない場合（木構造）でもグラフと呼ぶことは出来ます。



グラフ

- 先に紹介した木構造との違いは、枝の書き方にも表れています。このグラフの例では、枝は矢印ではなく単なる線になっています。この様な枝をもつグラフを無向グラフと呼びます。実はグラフの枝を矢印にすることも可能で、枝が矢印になっている（枝に向きがある）グラフを有向グラフと呼びます。
- 有向グラフは例えば一方通行を含む交通網や、SNSのフォロー関係（ユーザーを点、フォローを枝とみなす）などを表現するときに用いられます。また、先程の例では、山手線の外回り（時計回り。勿論、内回りでも構いません）に着目した場合、一部を有向グラフで表すことが出来ます。