

こう ぞう  
# データ構造とアルゴリズム  
なか かたづ さが かた  
### コンピュータの中の「お片付け」と「探し方」

# # プログラム

## ざいりょう かた 「材料のしまい方」



\* データ構造

\* 情報をきれいにコンピュータに  
しまう方法です。

## りょうり てじゆん 「料理の手順」



\* アルゴリズム

\* 情報を使って、計算したり  
探したりする順番です。

# # 変数と配列



## \* 1. 変数 (1つの箱)

- \* データを 1つだけ入れることができます。



## \* 2. 配列 (ロッカー)

- \* 同じ形のデータを並べて入れます。
- \* 添字 (ロッカーの番号) で場所を決めます。

# # リストりすと



- \* はな 離れた場所にあるデータをつなぐ方法です。
- \* **ポイント** (つぎ 次の場所を教える地図) を使います。
- \* 「宝探し」のように、たからさが 次から次へ進みます。

# # キューとスタック

## キュー (レジの<sup>じゅんばん</sup>順番)

さきい さきだ  
先入れ 先出し (FIFO)

[IN] → [Line] → [OUT]



さいしょ きひと さいしょで  
最初に来た人が、最初に出ます。

## スタック (お皿の<sup>さら やま</sup>山)

あとい さきだ  
後入れ 先出し (LIFO)

[IN/OUT] ↔ [Stack]



さいご おさら さいしょと  
最後に置いたお皿を、最初に取ります。

# # 木構造 (ファイルシステム)

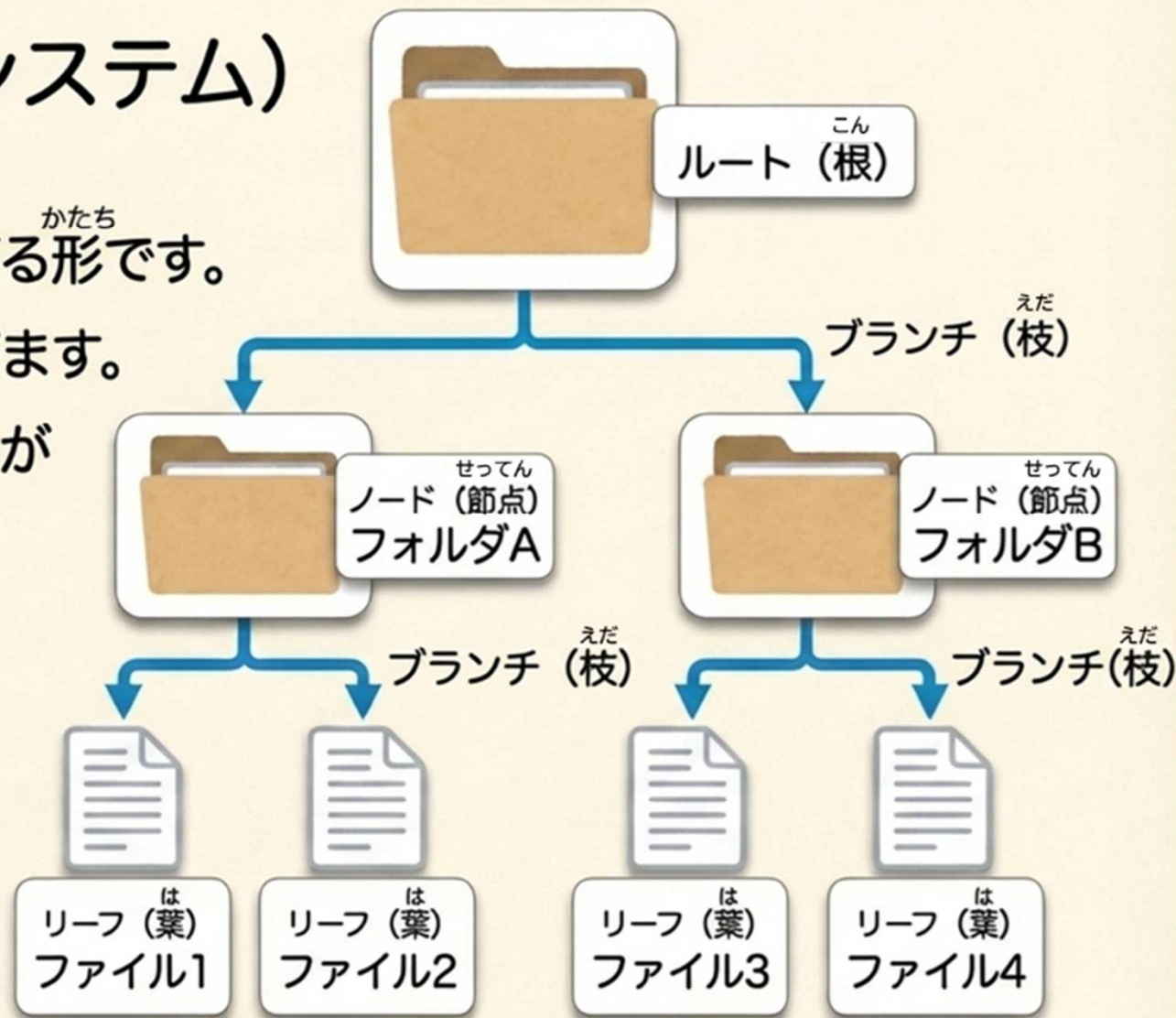
\* 親から子へ、上から下へつながる形です。

\* 各要素をノード (節点) と呼びます。

\* 特別なノードと、その間のつながりを以下のように呼びます：

- ルート (根) : 一番上のノード
- ブランチ (枝) : ノードをつなぐ線
- リーフ (葉) : 一番下のノード (子を持たない)

\* 枝が2つ以下の形を  
2分木 と言います。



# # アルゴリズムと流れ図

\* コンピュータに <sup>しごと</sup>仕事を  
させる手順です。

\* 手順が <sup>てじゅん</sup>少なく、<sup>すく</sup>わかりやすい  
ものが「良いアルゴリズム」です。

\* <sup>なが</sup>流れ図（フローチャート）を  
<sup>つか</sup>使って、<sup>え</sup>絵で<sup>あらわ</sup>表します。



# # ながす きほん 流れ図の3つの基本

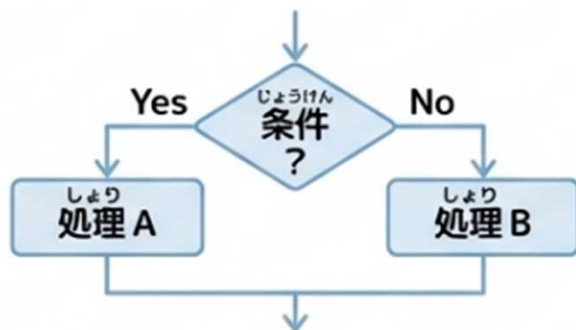
## 1. 順次構造

うえ した じゆんぱん うちご  
上から下へ、順番に動きます。



## 2. 選択構造

せんたくこうぞう  
「はい」か「いいえ」で、道が分かります。



## 3. 繰返し構造

くりかえ こうぞう  
おなじ こと を 何度も 繰り返します。(ループ)



# # 探索 (データをさがす)

## せんけいたんさく 線形探索 (1ページずつよ読む)

さいしょ じゅんばん さが  
最初から1つずつ順番に探します。  
さいご さが お ばんべいほう  
※最後に探すものを置く番兵法もあります



## にぶんたんさく 二分探索 (じしょのまなかからひらく)

ま なか くら はんぶん しほ こ  
真ん中のデータと比べて、半分に絞り込みます。  
なら  
※並んでいるデータだけ



# せいでつ なら # 整列 (データをきれいに並べる)

ちい じゅん しょうじゅん おお じゅん こうじゅん  
※小さい順 (昇順) や大きい順 (降順) にします。

きほんせんたくほう

## 基本選択法

いちばん ちい  
- 一番小さいものを選んで、先頭 (前) に動かします。



きほんこうかんほう

## 基本交換法 (バブルソート)

となり くら い か  
- 隣にある2つのデータを比べて、入れ替えます。



きほんそうにゆうほう

## 基本挿入法

と だ ただ ばしょ い  
- データを取り出して、正しい場所に入れます。



# まとめ

## データ構造 (しまい方)

## アルゴリズム (やり方)



### \* データ構造

- 変数、配列、リスト、キュー、
- スタック、木構造

### \* アルゴリズム

- 形：順次、選択、繰返し
- 探す：線形探索、二分探索
- 並べる：選択、交換 (バブル)、挿入