アルゴリズム入門はじめに

山口 泰 TA: 安達瑛翔, 王 楽佳 S1 – 1~5, 15

アルゴリズム入門で学ぶこと

- ・「情報」に関する科学的な基礎
 - 様々な分野に現われる「情報」を扱うための科学
 - cf. 数学は様々な分野の「数理」を、物理は「物」を扱う ための科学
 - ・生物の理解: 個体→分子→情報
 - ・ロボット工学: 機械→制御→ソフトウェア
 - 科目「情報」の土台部分

– 情報を扱う学問分野の入り口

授業計画

- ・ 第1部 必要最小限のプログラミング
 - 第1章 数の計算と関数
 - 第2章 配列による画像の表示
 - 第3章 条件分岐と繰り返し
 - 第4章 関数から計算へ
- 第2部 プログラミングを通して学ぶ情報科学の 諸概念
 - 第5章 アルゴリズムと計算量
 - 第6章 数値計算
 - 第7章 パターン認識
 - 第8章 レコードとオブジェクト (**)
 - 第9章 再帰データ構造 (*)
 - 第10章 いろいろなプログラミング言語 (*)

ご利益

- 情報システムが何ができるのか?が分かる
 - 「ゲノムデータベースから検索をするのにかかる 時間は?」
 - -- 「ロボットに滑らかに動きをさせるのが大変なの は何故?」

- 実際にやらせてみないと分からない!

情報科学の基礎知識がいくつか分かる

 アルゴリズム/数値計算/パターン認識

 プログラミングの入門ができる(副作用)

 言語の違いは大きな問題ではない!

授業形式·評価

- ・クラス指定
- ・講義(1/2~2/3)+演習(1/3~1/2)
- 講義はスライド・板書を用いる
- 演習はプログラミングを通して行う

- プログラミングが主目的ではない!

- レポート課題 概ね半分
- 期末試験 概ね半分

URL

lecture.ecc.u-tokyo.ac.jp/~yama/

• アルゴリズム入門共通資料

lecture.ecc.u-tokyo.ac.jp/johzu/joho-kagaku/ ここに多くの情報があります!

ITC-LMS

- ITC-LMSの利用登録 ウェブブラウザ(Safariなど)から https://itc-lms.ecc.u-tokyo.ac.jp/portal/login/ にログインして、「コース検索」から「0050156 アルゴ リズム入門」を選択する
- ITC-LMSの利用
 ログインして、「LMS」で「時間割」表に「アルゴリズ ム入門(月4)」があることを確認する



- 目的(成績には使わない)
 - 出席状況の確認
 - 授業の理解度の確認
 - 演習の進捗状況の確認
- 使い方
 - ITC-LMS(授業のWWWページ)から、
 - 投票クライアント c.rb を(ホームに)ダウンロード
 - 投票(ターミナル利用/カレントディレクトリに注意)

\$ cd

- \$ ruby c.rb サーバ名 アカウント名 選択肢番号
- 各問、同一の端末からは一回のみ投票可能。

科目「情報」は面白かったか

- 1. 面白かった
- 2. ふつう
- 3. つまらなかった

科目「情報」は難しかったか

- 1. 難しかった
- 2. ふつう
- 3. やさしかった



- 1. 自由にプログラムを書くことができる
- 2. プログラムをしたことがある
- 3. プログラムをしたことがない

数の計算と関数

プログラミング言語Ruby

- 概念を理解するためにプログラミング演習を行う
 手段であり目的ではない
 - 簡単に試せるのも情報関係の特徴
- プログラミング言語 Ruby を用いる
 - 正確には irb という対話型の処理系~簡単に試せる

 実際には極めて実用的な言語(だが、この授業では 深入りしない)
 - 産業界・科学技術研究開発で使われている他のプロ グラミング言語を学ぶ際の土台にもなる
 - 日本発のオープンソースソフトウェア

irbとisrb

- グラフィックス表示の機能 show を使うためには、irb の代わりに isrb(以前は isrb2 だった)を用いる。
- ただし、ECC の Mac OS X の端末の前でしか 使えない。

コンピュータとの対話

- ターミナルの起動 \Rightarrow irbの起動 \Rightarrow 数式の入力
 - ターミナルの pc123456m:~ 9474079989\$ irb 改行 プロンプト irbの
- プロンプト

irbの

返答

=> 2

- ▶irb(main):001:0> 1+1 改行
 - irb(main):002:0> コントロールD

pc123456m:~ 9474079989\$

• 今回は、指示されるまで、irbを使いながら…



電卓がわり

irb(main):009:0>7 - 2 * 3 => 1 irb(main):010:0> (7 - 2) * 3 => 15 irb(main):012:0> 7.0 / 2 => 3.5 irb(main):013:0>7/2.0 => 3.5

17 - 17/3*3の値は

- 1. 16
- 2. 15.111111111111
- 3. 2
- 4. 0.0
- 5. 0

56の16乗として間違っているのは

- 1. 56 ** 16
- 2. (7 * 8) ** 16
- 3. 7 * 8 ** 16
- 4. 56 ** 4 ** 2
- 5. 56 ** (4 ** 2)

さまざまなエラー irb(main):001:0> 3/0 ZeroDivisionError: divided by 0 from (irb):1:in `/' from (irb):1 irb(main):002:0>7-23 (irb):2: syntax error, unexpected tINTEGER, expecting Send 式の途中で改行すると プロンプトが異なる from (irb):2 irb(main):003:0> (7 irb(main):004:1*2)*3(irb):4: syntax error, unexpected ')', expecting \$end from (irb):4 irb(main):005:0>

さまざまなエラー

irb(main):013:0> bm1(188.0, 104.0)

NoMethodError: undefined method 'bm1' for main: Object from (irb):13

わけがわからなくなったら

ともかくコントロールCを押す
 – irb のトップレベルに戻る

数学関数

数学関数を使う準備

irbを起動し直す

たびに必要

- irb(main):003:0> include(Math)
- => Object
- irb(main):004:0> sqrt(2)
- ⇒1. 4142135623730951
- irb(main):005:0> cos(3.141592/3)
- ⇒0. 5000001886751098
- irb(main):005:0> cos(PI/3)
- => 0.50000000000001

フィボナッチ数列を知っている?

- 1. 知っている
- 2. 知らない

フィボナッチ数列の100番目の数値は

- 1. 偶数
- 2. 奇数

黄金比の値は

黄金比 φ とは, 1: φ = φ: (φ+1) となる値

- 1. 1.4142135623731
- 2. 1.41592653589793
- 3. 1.618033988749895
- 4. 1.61803398874985
- 5. 1.73205080756888

 $\phi = \frac{1 + \sqrt{5}}{2}$

BMIを知っている?

- 1. 知っている
- 2. 知らない



- => 104.0
- irb(main):006:0> w / (h /100.0) ** 2
- => 29.425079221367138

変数を使うわけ

body_weight_in_pound

・ 違う値で計算のやり直しができる

irb(main):008:0> w=104.0-10

w=w-10
としてもよい

=> 94.0

irb(main):009:0> w / (h/100.0) ** 2

=> 26.595744680851066

irbへの入力

- コントロールPもしくは上矢印を入力すると、直前の 入力が復活する。
- コントロールNもしくは下矢印を入力すると、次の入力が復活する。
- コントロールBもしくは左矢印でカーソルは左に移動。
- ・コントロールFもしくは右矢印でカーソルは右に移動。
- 通常の文字はカーソル位置に挿入される。
- ・コントロールDでカーソル位置の文字が削除される。
- デリートでカーソルの直前の文字が削除される。

関数の定義 --- BMIを求める関数

=> :bmi

irb(main):007:0> bmi(188.0, 104.0)

=> 29.425079221367138

irb(main):008:0> 1.1*bmi(174.0, 119.0 * 0.454)

=> 19.628947020742505

涷漝

- 1. 平面上の 2 点 (x, y) と (u, v) の距離を求める distance(x,y,u,v).
- 2. f フィートi インチをセンチメートルに変換する feet_to_cm(f,i). ただし、1 フィート = 12 イン チ = 30.48 cm である。
- p ポンド o オンスをキログラムに変換する pound_to_kg(p,o). 1 ポンド = 16 オンス = 0.4536 kg である。

進捗状況の確認

- 1. すべてできた
- 2. 二つできた
- 3. 一つできた

4. できない

解答

```
def distance(x, y, u, v)
  sqrt((x-u)^{*}(x-u) + (y-v)^{*}(y-v))
                      # sqrt((x-u)**2+(y-v)**2)
end
def feet to cm(f, i)
  30.48*f + 30.48/12*i # (f + i/12.0)*30.48
end
def pound to kg(p, o)
  0.4536^{*}p + 0.4536/16^{*}o
                          \# (p + o/16.0) * 0.4536
End
```

ここまで

- 1. 理解した
- 2. よくわからないところがある
- 3. 全然わからない

関数を使う関数

関数も、 小文字で始まる英数字列 アンダースコアは 小文字と考える

irb(main):010:0> def bmi_yp(f,i,p,o) irb(main):011:1> bmi(feet_to_cm(f,i), irb(main):012:2* pound_to_kg(p,o)) irb(main):013:1> end => nil 式の途中で改行すると プロンプトが異なる

irb(main):015:0> bmi_yp(5,11,170,0)

=> 23.710342996960538

1.6 定義のまとめ

この章で紹介した式や命令をまとておこう。

include(Math): cos や sqrt などの数学関数を使う前に、実行しておかな ければいけない命令。

| 変数名 | = | 式 | : 右辺の式を計算した値を、左辺に書かれた名前の変数にし まう代入命令。

関数名 (式₁ ,..., 式_n): 式₁ ,..., 式_n を計算した値を、関数に渡す関 **数呼び出し式**。関数に渡される式の値のことを引数という。 関数名 は sqrt のような予め定義されている関数でも、bmi のような自分で定義 した関数でもよい。 def <mark>関数名</mark> (変数名₁,...,変数名_n) \Box 式 end: 関数定義。少し 複雑なので、具体例で説明する。



def と end は、定義の範囲を示している。**関数の名前**(ここでは bmi)は、 変数名と同様、好きな名前を付けることができる。次の (height,weight) は、関数が値を受け取るために使う変数名である。

次回に向けて - Emacs

- Emacs (イーマックス) は歴史の長い代表的な
 テキスト・エディタである。
- コントロール・キーを使うのが由緒正しいが、
 とりあえず、メニューや矢印キーを使ってテキ
 ストを編集しよう。
- irb でロードする前にセーブを忘れずに。
- 自分の慣れたエディタがあれば、それを使ってもよいです。ただし、プレーン・テキストのファイルが作れないと駄目。

ディレクトリは?

- 1. 授業用のディレクトリで作業している
- 2. ホームディレクトリで作業している
- 3. デスクトップ(自体)で作業している
- 4. ホームディレクトリ・デスクトップって何?
- 5. 上記以外

パスは?

- 1. 絶対パス・相対パスともに理解している
- 2. 絶対パスがわからない
- 3. 相対パスがわからない
- 4. 絶対パスも相対パスも理解できていない

エディタは

- 1. Emacs を使ったことがある/ている
- 2. miを使ったことがある/ている
- 3. テキストエディットを使ったことがある/ている
- 4. その他のエディタを使ったことがある/ている
- 5. 使ったことがない
- 6. エディタって何?

レポート

・授業ページにしたがってメールする。

yama+is@lecture.ecc.u-tokyo.ac.jp

• Subject:欄は、

assignment-1

としてください。

ダ切: 10月 2日(月) 授業開始前
 – 今回は成績に加味しません。
 – 習得状況を調査するためです。