

# 目次

## #1: アルゴリズムとは/Python を動かしてみる/関数と計算

- はじめに, pp.1
- 1: コンピュータの特徴, pp.1
  - 1.1: アナログとデジタル, pp.1
  - 1.2: コンピュータとデジタル情報, pp.2
  - 1.3: モデル化とコンピュータ, pp.3
- 2: アルゴリズムの特徴, pp.4
  - 2.1: アルゴリズムとその記述方法, pp.4
  - 2.2: 変数と代入・手続き型計算モデル, pp.5
- 3: アルゴリズムとプログラミング言語, pp.5
  - 3.1: プログラミング言語, pp.5
  - 3.2: Python 言語による記述, pp.6
  - 3.3: 動かしてみよう!, pp.6
  - 3.4: 数値の表示に関する補足, pp.8

## #2: 数値表現と誤差についての考え方

- 注意: インデント (字下げ) について, pp.1
- 1: 前回の演習問題の解説, pp.2
  - 1.1: 演習 1-3a — 四則演算を試す, pp.2
  - 1.2: 演習 1-3b — 剰余演算, pp.3
  - 1.3: 演習 1-3c — 円錐の体積, pp.4
  - 1.4: 演習 1-3d — 四則演算の精度を調べる, pp.5
- 2: コンピュータ上での数値の表現, pp.6
  - 2.1: 十進表現と二進表現, pp.6
  - 2.2: 負数の表現と二の補数, pp.6
  - 2.3: 実数の表現と浮動小数点, pp.8
  - 2.4: 浮動小数点と誤差, pp.10

## #3: 制御構造 (枝分かれ/繰り返し)

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 2-1 — 整数と実数の違い, pp.1
  - 1.2: 演習 2-2 — 実数計算の誤差, pp.2
  - 1.3: 演習 2-3 — 誤差の種類, pp.3
  - 1.4: 演習 2-4 — 誤差の理由, pp.3
- 2: 基本的な制御構造, pp.3
- 3: 枝分かれと if 文, pp.4
- 4: 繰り返しと while 文, pp.7
- 5: 制御構造の組み合わせ, pp.9
- 付録: 構造化定理の補足, pp.11

## #4: 数値解析の基本

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 3-1a — 枝分かれの復習, pp.1
  - 1.2: 演習 3-1b — 枝分かれの入れ子, pp.2
  - 1.3: 演習 3-1c — 多方向の枝分かれ, pp.4
  - 1.4: 演習 3-2 — 誤差の補正, pp.5
  - 1.5: 演習 3-3 — 計算の繰り返し, pp.6
  - 1.6: 演習 3-4a — 条件の組み合わせ, pp.6
  - 1.7: 演習 3-4b — 条件の排他性, pp.7
  - 1.8: 演習 3-4c — フラグを用いた条件, pp.8
- 2: 数値積分, pp.9
- 3: 計数ループ, pp.11
- 4: 微分方程式の数値解法, pp.12
  - 4.1: 微分方程式, pp.12
  - 4.2: オイラー法, pp.13
  - 4.3: ルンゲ-クッタ法, pp.15
- 付録:  $dx$  と  $dy$  について, pp.17

## #5: 差分法/差分法の誤差/計算安定性についての考え方 (1)

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 4-1a~4-1c — 数値積分, pp.1
  - 1.2: 演習 4-2a~4-2c — 繰り返し, pp.3
- 2: 差分法, pp.4
  - 2.1: 差分方程式, pp.4
  - 2.2: テイラー級数, pp.5
  - 2.3: オイラー法の誤差, pp.5
  - 2.4: ルンゲ-クッタ法の誤差, pp.6
- 3: 数値計算の安定性, pp.7
- 付録 1: ラグランジュの補間式について (副題: 意味を考えよう), pp.10
- 付録 2: 差分法の安定化について (教科書の補足), pp.14

## #6: 計算安定性についての考え方 (2)/データ構造と配列

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 5-1 — テイラー級数で  $\sin$  と  $\cos$  を計算, pp.1
  - 1.2: 演習 5-2b — ロジスティック方程式の差分, pp.2
  - 1.3: 演習 5-3 — 素数発見と実行時間の計測, pp.8
- 2: 様々なデータ型, pp.9
  - 2.1: 基本データ型, pp.9
  - 2.2: 構造データ型・複合データ型, pp.10
  - 2.3: Python における複合データ型, pp.11
  - 2.4: 配列の生成とその処理, pp.12
  - 2.5: 配列の利用, pp.14

## #7: 2次元配列/セル オートマトンによるシミュレーション

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 6-1 — 配列の生成, pp.1
  - 1.2: 演習 6-2 — 配列の取り扱い, pp.2
  - 1.3: 演習 6-3 — 素数発見方法の改良, pp.3
  - 1.4: 演習 6-3 — 素数発見方法の改良, pp.6
- 2: 2次元配列, pp.8
- 3: セル オートマトン, pp.9
  - 3.1: 1次元セル オートマトン, pp.9
  - 3.2: セル オートマトンの応用, pp.13
- 参考: セル オートマトンを用いて砂漠化の進行をシミュレートする Python プログラム, pp.17

## #8: 相関係数/回帰分析/仮説検定/連立 1 次方程式の数値的解法

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 7-1 — 2次元配列の生成, pp.1
- 2: 未知データの傾向分析, pp.2
  - 2.1: 相関係数, pp.2
  - 2.2: 回帰分析の数値解法, pp.4
  - 2.3: 仮説検定, pp.6
- 3: 連立 1 次方程式の数値的解法, pp.8
- 付録 1: 線形回帰分析と誤差の二乗和との関係, pp.15
- 付録 2: 母集団と観測データとの関係, pp.18

## #9: ランダム アルゴリズム/再帰呼び出し

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 8-2a — p 値を用いた仮説検定, pp.1
  - 1.2: 演習 8-2b — 十分な仮説検定に必要なデータ数 (自由度), pp.2
- 2: 乱数とランダム アルゴリズム, pp.4
  - 2.1: 乱数とは, pp.4
  - 2.2: 擬似乱数, pp.4
  - 2.3: ランダム アルゴリズム, pp.5
  - 2.4: モンテカルロ法, pp.5
  - 2.5: モンテカルロ法の誤差, pp.6
- 3: 再帰呼び出し, pp.10
  - 3.1: 再帰手続き・再帰関数の考え方, pp.10
  - 3.2: 再帰呼び出しの興味深い特性, pp.11
  - 3.3: 再帰呼び出しによる枝分かれ, pp.12
  - 3.4: 再帰呼び出しによる順列の列挙, pp.16

## #10: 整列アルゴリズムとその所要時間

- 再帰処理の考え方, pp.1
- 再帰処理における副作用について, pp.1
- 1: 前回の演習問題の解説, pp.2
  - 1.1: 演習 9-2: — 再帰関数, pp.2
  - 1.2: 演習 9-3a: — 単純減少列の列挙, pp.3
  - 1.3: 演習 9-3b: — 重複を許す順列の列挙 1, pp.4
  - 1.4: 演習 9-3c: — 重複を許す順列の列挙 2, pp.5
- 2: 整列アルゴリズム, pp.6
  - 2.1: 単純選択法, pp.6
  - 2.2: 単純挿入法, pp.7
  - 2.3: バブル ソート, pp.9
  - 2.4: マージ ソート, pp.10
  - 2.5: クイック ソート, pp.13
  - 2.6: 整列アルゴリズムの拡張と検索, pp.14
- 3: 整列アルゴリズムの計測, pp.15

## #11: 計算量の考え方

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 10-1: — 各整列アルゴリズムの所要時間, pp.1
    - \* 1.1.1: 単純選択法の計算量, pp.1
    - \* 1.1.2: マージ ソートの計算量, pp.2
    - \* 1.1.3: クイック ソートの計算量, pp.3
  - 1.2: 演習 10-2 — クイック ソートの弱点, pp.5
- 2: 時間計算量, pp.6
  - 2.1: 時間計算量の考え方, pp.6
  - 2.2: 時間計算量の例, pp.7
- 3: 時間計算量の見積もり方, pp.8
  - 3.1: 整列アルゴリズムの時間計算量, pp.8
  - 3.2: 最大公約数, pp.8
  - 3.3: フィボナッチ数, pp.9
  - 3.4: 組み合わせ数, pp.11

## #12: 動的計画法とトレース バック

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 11-1a ~ 11-1h: — 様々なメソッドの計算量, pp.1
  - 1.2: 演習 11-2 — ユークリッドの互除法の計算量, pp.2
- 2: 動的計画法, pp.3
  - 2.1: 動的計画法とは, pp.3
  - 2.2: 部屋割り問題 (1次元配列を利用する例), pp.4
  - 2.3: 経路数問題 (2次元配列を利用する例), pp.7
- おまけ: ゲームの作成, pp.13

## #13: パターン認識/クラスタリング/計算複雑性

- 1: 前回の演習問題の解説, pp.1
  - 1.1: 演習 12-1: — 部屋割り問題の拡張, pp.1
  - 1.2: 演習 12-2: — 釣り銭問題, pp.2
  - 1.3: 演習 12-3: — 経路数算出問題の拡張, pp.2
- 2: パターン認識, pp.4
  - 2.1: パターン認識とは, pp.4
  - 2.2: 文字列の対応付け, pp.4
  - 2.3: 参考: 遺伝子のアラインメント, pp.10
- 3: クラスタリング, pp.14
  - 3.1: クラスタリングとは, pp.14
  - 3.2: k-means 法, pp.14
- 最後に, pp.18
- 参考: k-means 法の Python プログラム, pp.19
- 参考: 組み合わせ集合を求める Python プログラム, pp.22