

索引

あ

- 値と参照の区分 #6: 2.2 構造データ型・複合データ型, pp.11
- 与えられたプログラムが有限時間に停止するかどうかを判断できるプログラム
..... #1: 2.1 アルゴリズムとその記述方法, pp.4
- アナログ量 (Analog Quantity) #1: 1.1 アナログとデジタル, pp.1
- 溢れ (Overflow) #2: 2.2 負数の表現と二の補数, pp.7
..... #3: 1.2 演習 2-2 — 実数計算の誤差, pp.2
- アルゴリズム (Algorithm) #1: はじめに, pp.1
..... #1: 2.1 アルゴリズムとその記述方法, pp.4
- 意味を考える #5: 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
..... (未知データを分析する最初の一步) #8: 2.1 相関係数, pp.2
- インデント (Indent) #2: 注意: インデント (字下げ) について, pp.1
- 打ち切り誤差 (Cutoff Error) #6: 1.1 演習 5-1 — テイラー級数で \sin と \cos を計算, pp.1
- 枝分かれの入れ子・枝分かれの並び #4: 1.3 演習 3-1c — 多方向の枝分かれ, pp.4
- エラトステネスのふるい #7: 1.4 演習 6-4 — 配列を用いた素数発見方法の改良, pp.7
- オイラー法 (Euler Method) #4: 4.2 オイラー法, pp.13
- オイラー法と差分法との関係 #5: 2.1 差分方程式, pp.4
- オイラー法の誤差 #5: 2.3 オイラー法の誤差, pp.5

か

- 解析的に解く #4: 2 数値積分, pp.9
- 外部整列 (External Sorting) #10: 2.4 マージソート, pp.12
- ガウスの消去法 (Gaussian Elimination) #8: 3 連立 1 次方程式の数値的解法, pp.8
- ガウス-ジョルダンの消去法 (Gauss-Jordan Elimination) .. #8: 3 連立 1 次方程式の数値的解法, pp.9
- カウンタ (Counter) #4: 3 計数ループ, pp.11
- カオス (Chaos) #6: 1.2 演習 5-2b — ロジスティック方程式の差分, pp.6
- 過去に作成したプログラムを新規プログラムの部品として使う
..... #4: 1.2 演習 3-1b — 枝分かれの入れ子, pp.3
- 仮数 (Mantissa) #2: 2.3 実数の表現と浮動小数点, pp.9
- 仮説検定 (Hypothesis Testing) #8: 2.3 仮説検定, pp.6
- 関数 (Function) #1: 3.2 Python 言語による記述, pp.6
- 擬似コード (Pseudo Code) #1: 2.1 アルゴリズムとその記述方法, pp.5
- 基数 (Radix) #2: 2.1 十進表現と二進表現, pp.6
- 基本データ型 (Primitive Data Types) #6: 2.1 基本データ型, pp.9
- 局所的最適解 (Local Optimal Solution) #13: 3.2 k-means 法, pp.17
- クイックソート (Quick Sort) #10: 2.5 クイックソート, pp.13

- クイックソートにおけるピボット選択 #10: 2.5 クイックソート, pp.13
- クイックソートの改良 #11: 1.2 演習 10-2 — クイックソートの弱点, pp.5
- クラスタリング (Clustering) #13: 3.1 クラスタリングとは, pp.14
- 計算量 (Computational Complexity) ... #11: 1.1 演習 10-1: — 各整列アルゴリズムの所要時間, pp.1
..... #11: 2.1 時間計算量の考え方, pp.6
- 計算量: クイックソート #11: 1.1.3 クイックソートの計算量, pp.3
- 計算量: 再帰とループの組合せ #12: 1.1 演習 11-1h, pp.2
- 計算量: 単純選択法 #11: 1.1.1 単純選択法の計算量, pp.1
- 計算量: 最大公約数の算出 #11: 3.2 最大公約数, pp.8
..... #12: 1.2 演習 11-2 — ユークリッドの互除法の計算量, pp.2
- 計算量: フィボナッチ数 #11: 3.3 フィボナッチ数, pp.9
- 計算量: べき乗計算 #12: 1.1 演習 11-1f, pp.1
- 計算量: マージソート #11: 1.1.2 マージソートの計算量, pp.2
- 計算量算出の一般的方法: 下線・太字部分を理解すること
..... #12: 1.2 演習 11-2 — ユークリッドの互除法の計算量, pp.3
- 組合せ数の計算 #5: 1.2 演習 4-2a~4-2c — 繰り返し, pp.4
..... #11: 3.4 組み合わせ数, pp.11
- クロネッカーのデルタ (Kronecker Delta)
..... #5: 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.12
- 計数ループ (Counting Loop) #4: 3 計数ループ, pp.11
- 決定的アルゴリズム (Deterministic Algorithm) #9: 2.3 ランダムアルゴリズム, pp.5
- 桁落ち (Cancellation) #2: 2.4 浮動小数点と誤差, pp.10
- 桁落ちの回避 #2: 2.4 浮動小数点と誤差, pp.11
- 格子点を用いた面積の計算 #9: 2.5 モンテカルロ法の誤差, pp.7
- 構造化定理 (Structure Theorem) #3: 2 基本的な制御構造, pp.4
..... #3: 付録: 構造化定理の補足, pp.11
- 構造化データ型 (Structured Data Type) #6: 2.2 構造化データ型・複合データ型, pp.10
- 誤差の種類・理由・補正 #3: 1.3 演習 2-3 — 誤差の種類, pp.3
..... #3: 1.4 演習 2-4 — 誤差の理由, pp.3
..... #4: 1.4 演習 3-2 — 誤差の補正, pp.5
..... #6: 1.1 演習 5-1 — テイラー級数で sin と cos を計算, pp.2
..... #6: 脚注 1, pp.1
- コメント (Comment) #4: 2 数値積分, pp.10
- コメントアウト (Comment Out)・コメントアウトを外す #4: 2 数値積分, pp.10
..... #4: 脚注 6, pp.10
- コード (Code) #1: 2.1 アルゴリズムとその記述方法, pp.5

ル

- 再帰 (Recursion) #9: 3.1 再帰手続き・再帰関数の考え方, pp.10
- 再帰呼び出しにおける“担当部分の記録”と“副作用の解消”の対
..... #10: 1.3 演習 9-3b: — 重複を許す順列の列挙 1, pp.5
- 再帰呼び出しの考え方 #10: 再帰処理の考え方, pp.1
- 再帰呼び出しの樹形図 #9: 3.3 再帰呼び出しによる枝分かれ, pp.14
- 再帰呼び出しのデバッグ #9: 3.3 再帰呼び出しによる枝分かれ, pp.15

- 再帰呼び出しの副作用 #9: 3.3 再帰呼び出しによる枝分かれ, pp.13
..... #10: 再帰処理における副作用について, pp.1
- 再帰呼び出しを用いたべき乗計算 #11: 脚注 8, pp.14
- 最小二乗法 (Least Squares Method) #8: 2.2 回帰分析の数値解法, pp.5
- サブルーチン (Subroutine) #1: 脚注 9, pp.6
- 差分方程式 (Difference Equation) #5: 2.1 差分方程式, pp.4
- 自然対数の底 (Base of Natural Logarithm) #2: 脚注 3, pp.4
- 順次実行 or 接続 (Sequencing) #3: 2 基本的な制御構造, pp.4
- 漸近法 #5: 1.1 演習 4-1a~4-1c — 数値積分, pp.3
- 辞書型 #6: 2.3 Python における複合データ型, pp.11
- シミュレーション (Simulation) #7: はじめに, pp.1
- 条件式・文 #3: 3 枝分かれと if 文, pp.5
- 条件の順番・条件の排他性 #4: 1.7 演習 3-4b — 条件の排他性, pp.7
- 仕様の確認 #4: 1.1 演習 3-1a — 枝分かれの復習, pp.2
- 情報落ち (Loss of Information) #2: 2.4 浮動小数点と誤差, pp.10
- 初期値鋭敏性 (Sensitivity to Initial Conditions)
..... #6: 1.2 演習 5-2b — ロジスティック方程式の差分, pp.6
- 所要時間の計測 #5: 演習 5-3, pp.9
..... #10: 3 整列アルゴリズムの計測, pp.15
- シンプソンの公式 (Simpson Rule) #5: 1.1 演習 4-1a~4-1c — 数値積分, pp.2
- 数値以外の整列 #10: 2.6 整列アルゴリズムの拡張と検索, pp.14
- 数値的に解く #4: 2 数値積分, pp.9
- ステップ幅 Δx と数値計算の安定性 #5: 3 数値計算の安定性, pp.9
..... #6: 1.2 演習 5-2b — ロジスティック方程式の差分, pp.4
- 制御構造 (Control Structure) #3: 2 基本的な制御構造, pp.4
- セル オートマトン (Cellular Automaton) #7: 3 セル オートマトン, pp.9
- セル オートマトンの応用例 #7: 3.2 セル オートマトンの応用, pp.13
- 線形回帰分析 (Linear Regression Analysis) #8: 2.2 回帰分析の数値解法, pp.5
- 添字番号 (Index) #6: 2.4 配列の生成とその処理, pp.12
- 素数発見方法とその改良: for ループの境界・配列の利用
..... #7: 1.3 演習 6-3 — 素数発見方法の改良, pp.3
..... #7: 1.4 演習 6-4 — 配列を用いた素数発見方法の改良, pp.6
- ソース コード (Source Code) #1: 3.3 動かしてみよう!, pp.6

た

- 大域的最適解 (Global Optimal Solution) #13: 3.2 k-means 法, pp.17
- 台形公式 (Trapezoid Rule) #5: 1.1 演習 4-1a~4-1c — 数値積分, pp.1
- 代入 (Assignment) #1: 2.2 変数と代入・手続き型計算モデル, pp.5
- 多項式時間 (Polynomial Time Complexity) #11: 2.1 時間計算量の考え方, pp.7
- タプル (Tuple) #2: 1.1 演習 1-3a — 四則演算を試す, pp.2
- 単純選択法 (Selection Sort) #10: 2.1 単純選択法, pp.6
- 単純挿入法 (Insertion Sort) #10: 2.2 単純挿入法, pp.7

- 逐次細分最適化法: 本講義内だけの名称 #4: 2 数値積分, pp.9
..... #4: 脚注 5, pp.9
- 抽象化 (Abstraction) #1: 1.3 モデル化とコンピュータ, pp.3
- チューリング機械 (Turing Machine) #7: 脚注 6, pp.12
- デジタル量 (Digital Quantity) #1: 1.1 アナログとデジタル, pp.1
- テイラー展開 #5: 2.2 テイラー級数, pp.5
- 手続き型計算モデル (Procedural Computational Model)
..... #1: 2.2 変数と代入・手続き型計算モデル, pp.5
- 手順・手続き (Procedure) #1: 1.3 モデル化とコンピュータ, pp.3
..... #1: 脚注 9, pp.6
- デバッグの基本方法: 右記の箇条書きを参考に #7: 3.2 セル オートマトンの応用, pp.15
- テント写像: 初期値鋭敏性 or カオスが生じる理由
..... #6: 1.2 演習 5-2b — ロジスティック方程式の差分, pp.7
- データ型 (Data Types) #6: 2.1 基本データ型, pp.9
- データの収集数とモデルの誤差 #5: 参考, pp.3
- 動的計画法 (Dynamic Programming) #12: 2.1 動的計画法とは, pp.4
- 動的計画法の基本: この事例で適用方法の基本を覚えよう
..... #12: 2.2 部屋割り問題 (1 次元配列を利用する例), pp.4
..... #13: 2.2 文字列の対応付け, pp.5
- 動的計画法の制約 #12: 脚注 3, pp.4
..... #13: 2.3 参考: 遺伝子のアラインメント, pp.14
- トレース バック情報 #12: 2.2 部屋割り問題 (1 次元配列を利用する例), pp.6

な

- 内包表記 (Comprehension) #8: 1.1 演習 7-1 — 2 次元配列の生成, pp.1
- 二進法 (Binary System) #2: 2.1 十進表現と二進表現, pp.6
- 二の補数表現 (Two's Complement Representation) #2: 2.2 負数の表現と二の補数, pp.6
- 二分探索 (Binary Search) #10: 2.6 バブル ソート, pp.14

は

- 配列: 1 次元・2 次元 #6: 2.4 配列の生成とその処理, pp.12
..... #7: 2 次元配列, pp.8
- 配列型と辞書型の使い分け #6: 脚注 15, pp.12
- 配列の処理: 生成・要素へのアクセス・要素の追加・要素の削除・要素数の取得
..... #6: 2.4 配列の生成とその処理, pp.12
..... #7: 2 次元配列, pp.8
- バブル ソート (Bubble Sort) #10: 2.3 バブル ソート, pp.9
- パラメタ (Parameter) #1: 3.2 Python 言語による記述, pp.6
- 引数 (Argument) #1: 3.2 Python 言語による記述, pp.6
- ビット (Bit) #1: 1.2 コンピュータとデジタル情報, pp.2
..... #1: 脚注 2, pp.2
- 複合データ型 (Compound Data Type) #6: 2.2 構造データ型・複合データ型, pp.10

- 複数の戻り値 #2: 1.1 演習 1-3a — 四則演算を試す, pp.2
- 符号反転 (Negation) #2: 2.2 負数の表現と二の補数, pp.7
- 浮動小数点 (Floating Point) #2: 2.3 実数の表現と浮動小数点, pp.9
- 負の除数による剰余演算 #2: 1.2 演習 1-3b — 剰余演算, pp.4
- 不偏推定量 #8: 付録 2:母集団と観測データとの関係, pp.18
- フラグ (Flag) #3: 4 繰り返しと while 文, pp.8
- フラグと break の利用 #3: 5 制御構造の組み合わせ, pp.10
..... #4: 1.8 演習 3-4c — フラグを用いた条件, pp.8
- プログラム (Program) #1: はじめに, pp.1
..... #1: 2.1 アルゴリズムとその記述方法, pp.4
- プログラミング言語 (Programming Language) #1: 3.1 プログラミング言語, pp.5
- 文 (Statement) #1: 3.2 Python 言語による記述, pp.6
- 平方根 (Square Root) #2: 脚注 8, pp.11
- べき乗 (Power) #2: 1.3 演習 1-3c — 円錐の体積, pp.4
- 変数 (Variable) #1: 2.2 変数と代入・手続き型計算モデル, pp.5
- 母集団 (Population) #8: 2.2 回帰分析の数値解法, pp.4

ま

- 丸め誤差 (Round-off Error) #2: 2.4 浮動小数点と誤差, pp.10
- マージ ソート (Merge Sort) #10: 2.4 マージ ソート, pp.10
- 無限大 (Infinity) #4: 1.5 演習 3-3 — 計算の繰り返し, pp.6
- メモ化 (Memoization) #12: 2.1 動的計画法とは, pp.4
- 文字列型 #6: 2.3 Python における複合データ型, pp.12
- モジュール化とその利点 #9: 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.3
- モデル (Model) #1: 1.3 モデル化とコンピュータ, pp.3
- モンテカルロ法 (Monte Carlo Method) #9: 2.4 モンテカルロ法, pp.5
- モンテカルロ法を用いた面積の計算 #9: 2.4 モンテカルロ法, pp.6

ら

- ラグランジュの補間式 #5: 1.1 演習 4-1a~4-1c — 数値積分, pp.3
- 乱数値 (Random Number) #9: 2.1 乱数とは, pp.4
- ランダム アルゴリズム (Randomized Algorithm) #9: 2.3 ランダム アルゴリズム, pp.5
- 連立方程式の数値的解法におけるピボット選択 #8: 3 連立 1 次方程式の数値的解法, pp.11
- 論理演算 (Logical Operation) #3: 3 枝分かれと if 文, pp.5

わ

- 割り算とその逆数の掛け算が常に同じ結果 (数値) となる場合
..... #3: 1.2 演習 2-2 — 実数計算の誤差, pp.2

A

- abs 関数 (Abs Function) #7: 1.1 演習 6-1 — 配列の生成, pp.2

B

- break 文 (Break Statement) #3: 5 制御構造の組み合わせ, pp.10
- break 文, continue 文, pass 文の違い #12: 脚注 7, pp.9

C

- continue 文 (Continue Statement) #12: 脚注 7, pp.9

D

- dx と dy について #4: 付録:構造化定理の補足, pp.17

E

- $e \pm$ 指数 #2: 2.3 実数の表現と浮動小数点, pp.9

F

- False: 偽 #3: 3 枝分かれと if 文, pp.5
- float 関数 (Float Function) #2: 2.3 実数の表現と浮動小数点, pp.8
..... #3: 1.1 演習 2-1 — 整数と実数の違い, pp.2
- for 文・ループ (For Statement・Loop) #4: 3 計数ループ, pp.11
..... #4: 脚注 8, pp.11

I

- if 文 (If Statement) #3: 3 枝分かれと if 文, pp.4
- int 関数 (Int Function) #2: 2.3 実数の表現と浮動小数点, pp.8
..... #3: 1.1 演習 2-1 — 整数と実数の違い, pp.2

M

- math ライブラリ (Math Library) #2: 脚注 3, pp.4

N

- NP 困難 (NP-hard) #13: 3.2 最後に, pp.18
- NP 問題・NP クラス (Non-deterministic Polynomial Time) #13: 3.2 最後に, pp.18
- NaN: Not a Number #2: 2.4 浮動小数点と誤差, pp.11
- None #2: 1.1 演習 1-3a — 四則演算を試す, pp.2

O

- OS: オペレーティング システム (Operating System) #9: 2.2 擬似乱数, pp.4

P

- P 問題・P クラス (Polynomial Time) #13: 3.2 最後に, pp.18
- p 値: 有意確率 (P-Value) #8: 2 仮説検定, pp.7
- pass 文 (Pass Statement) #4: 1.6 演習 3-4a — 条件の組み合わせ, pp.7
- π : 円周率 (π : Circle Ratio:) #2: 1.3 演習 1-3c — 円錐の体積, pp.4
..... #2: 脚注 3, pp.4
- pprint 関数 (Pprint Function) #7: 2 2 次元配列, pp.9
- print 関数 (Print Function) #1: 3.4 数値の表示に関する補足, pp.8
- print 関数の書式化文字列 (Format Specifiers of Print Function)
..... #1: 3.4 数値の表示に関する補足, pp.8
- print 関数の end オプション #4: 1.7 演習 3-4b — 条件の排他性, pp.7
- Python における数値演算の結果: 整数・実数 #2: 2.3 実数の表現と浮動小数点, pp.8
..... #3: 1.1 演習 2-1 — 整数と実数の違い, pp.1
- Python における整数の範囲 #2: 2.2 負数の表現と二の補数, pp.8
- Python における複合データ型 #6: 2.3 Python における複合データ型, pp.11

R

- random.random 関数 (Random.random Function) #9: 2.2 擬似乱数, pp.4
..... #9: 脚注 5, pp.4
- range 関数 (Range Function) #6: 1.3 演習 5-3 — 素数発見と実行時間の計測, pp.8
..... #10: 2.2 単純挿入法, pp.8
- return 文 (Return Statement) #1: 3.2 Python 言語による記述, pp.6
- return の様々な使い方 #2: 1.1 演習 1-3a — 四則演算を試す, pp.2

S

- subleq 命令 #7: 3.1 1 次元セル オートマトン, pp.11

T

- True: 真 #3: 3 枝分かれと if 文, pp.5
- type 関数 (Type Function) #6: 2.3 Python における複合データ型, pp.11

W

- while 文・ループ (While Statement・Loop) #3: 4 繰り返しと while 文, pp.7