

Logical Thinking and Programming

論理思考とプログラミング

第5回－1時限目

アルゴリズム概念の理解と構築

Logical Thinking

Logical Thinking

今日の授業の流れ

- 1時限目
 - 講義
 - 1限目の演習範囲の解説
 - 並び替えアルゴリズムについて、指定問題の取り組み方
 - 演習
 - Project10 並び替えをしてみよう<前半 10.1~10.2>
- 2時限目
 - 講義
 - まとめ: 最小値選択法の効率計算
 - 2限目の演習範囲の解説、指定問題の取り組み方とポイント
 - まとめ: 制御構造の組み合わせ(復習)と最小値選択法
 - 演習
 - Project10 並び替えをしてみよう<後半 10.3~10.4>

Logical Thinking and Programming

Logical Thinking

配布資料

- 1人1枚
 - 第5回 演習チェックシート
- ペアで1セット
 - カード
 - 1セット40枚
 - 他のクラスでも使用するため、紛失ないように
 - 最小値選択法 実験用台紙(A3)
 - 最小値選択法 ワークシート(A4×4枚)
 - グラフ作成用エクセルテンプレート(授業のページからダウンロードできる)

Logical Thinking and Programming

Logical Thinking

1時限目の目標

- 与えられた(並び替え)アルゴリズムを手作業で実行することによって理解する

Logical Thinking and Programming

Logical Thinking

1時限目の演習範囲

- Project10 並び替えをしてみよう<前半 10.1~10.2>
 - テキストの範囲
 - P.117~P.122
 - 指定問題
 - やってみよう No.10-1(P.122)

Logical Thinking and Programming

Logical Thinking

並び替えアルゴリズム

- たくさんの要素を小さい順(昇順)あるいは大きい順(降順)に並べ替える「作業の手順(アルゴリズム)」
 - 例: 複数のカードをその番号を基準に、昇順に並べ替える

22	96	1	78	13
----	----	---	----	----

↓

1	13	22	78	96
---	----	----	----	----

Logical Thinking and Programming

Logical Thinking

「やってみよう No.10-1」について

- 概要
 - 40枚のカードを手作業で昇順に並び変えます
 - 最小値選択法(選択ソート)という方法で、並び替えを行います
 - 2人組(並び替え係、時間の記録係)のペアで行います
- 注意
 - まず、テキストP.117~P.122をよく読む
 - テキストの付録C「最小値選択法 ワークシート」を印刷したものを配布するので、それに従って演習を行うこと

Logical Thinking and Programming

Logical Thinking

「やってみよう No.10-1」の取り組み方1

- ペアを組む
 - ペアが組めない場合は、SAやTAとペアを組み、SAやTAが時間計測係を担当する
- ワークシートにペアの氏名を記入する
- ワークシートのStep1.とStep2.を読んで、カードの準備をし、並び替えアルゴリズムを理解する
 - カードをよくシャッフルすること
 - アルゴリズムはペアの両人が理解しておくこと
 - 5枚程度のカードでまず試してみるとよい
 - 同一の番号が振られたカードが複数枚あるが、Step2.の手順に従えば、正しく並び替えが行える

Logical Thinking and Programming

Logical Thinking

「やってみよう No.10-1」の取り組み方2

- カードの並び替えにかかる時間を計測し、結果をStep3.に記入していく
 - 並び替え係、時間計測係で分担する
 - 時間計測係はストップウォッチを作成する。テキスト P.121~122にストップウォッチの作成方法が掲載されている
- Step4.~Step7.へ解答を記入する
 - 2人で相談しながら記入してよい
 - Step4.で使うエクセルのテンプレートは授業のページからダウンロードできる
- ワークシートは2時限目の授業終了後に演習チェックシートと一緒に提出する
- カードが40枚あるか確認し、輪ゴムでとめて返却する

Logical Thinking and Programming

Logical Thinking and Programming

論理思考とプログラミング

第5回 - 2時限目

アルゴリズム概念の理解と構築

Logical Thinking

Logical Thinking

2時限目の目標

- 手作業で理解した(並び替え)アルゴリズムをプログラムとして書けるようになる

Logical Thinking and Programming

Logical Thinking

まとめ: 最小値選択法の効率計算


- 並び替えるカードの枚数と比較回数
 - カードが1枚の場合→0回
 - カードが2枚の場合→1回
 - カードが3枚の場合→2+1回
 - カードが4枚の場合→3+2+1回
 - カードが5枚の場合→4+3+2+1回
- カードの枚数がN枚の場合の比較回数
 - $(N-1)+(N-2)+(N-3)+\dots+1=N \times (N-1)/2$
 - 比較回数はNの2乗に比例して増加する
 - 計算結果は $(N^2-N)/2$ であるが、Nが十分に大きければ、 $N/2$ は無視できる
 - カードの枚数が2倍になると、時間は4倍かかる

Logical Thinking and Programming

講師によるデモ

Logical Thinking

- 第5回 > Project10 > 10.3~10.4



第5回 > Project10 > 10.3~10.4

Logical Thinking and Programming

2時限目の演習範囲

Logical Thinking

- Project10 並び替えをしてみよう < 後半 10.3 ~ 10.4 >
 - テキストの範囲
 - P.123~P.130
 - 指定問題
 - やってみよう No.10-2 (P.127)
 - やってみよう No.10-3 (P.130)
 - 発展問題
 - やってみよう No.10-4 (P.130)

Logical Thinking and Programming

やってみよう No.10-2

Logical Thinking

- 概要
 - 1限目に手作業で実行した最小値選択法をコンピュータで再現します
 - 2人組は解散し、1人で作業をします
- 注意
 - テキスト P.123~P.130をよく読む
 - 入れ物の使い方に関しては、テキストのP.163「付録 A.17」を参照すること

Logical Thinking and Programming

指定問題の取り組み方

Logical Thinking

- P.123~P.127までを読んで、カードの準備をし、「やってみよう No.10-2」(P.127)に取り組む
- P.128~P.130までを読んで、途中まで実装をして、残りを「やってみよう No.10-3」(P.130)として取り組む

Logical Thinking and Programming

実装時に気をつけるポイント

Logical Thinking

- タイルをよく読み、命令に間違いがないか丁寧に確認する
 - 例: 追加先の入れ物を間違える
 - 「●●に××に追加する」という命令のつもりで、「●●を△△に追加する」という命令を入れてしまう
- フローチャートを確認しながら実装する
 - フローチャートを無視して、タイルをいじくりまわすだけでは無理です
- 1回だけ実行ボタン **実行** を有効活用する
 - 1回実行してうまく行かないものは、テクタク(繰り返し)をしても、絶対にうまくいかない
 - 1回だけ実行して、命令がうまく動いていることを確認しながら、少しずつ作っていくことが大切

Logical Thinking and Programming

実装方法について

Logical Thinking

- 良い方法
 1. 少し作ったら、テストをする
 2. テストをしてうまくいかなかったら、一番最近追加した部分が間違っていることが分かる
 3. 間違いを修正し、意図通りに動いたら1.に戻る(完成まで繰り返し)
- 悪い方法
 1. 一度も実行(テスト)をしないで、一気に完成させる
 2. 実行すると、意図通りに動かない
 3. どこが間違っているか、正しいか分からない
 4. 手当たり次第に変更し、混乱してしまう

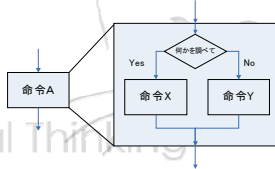
Logical Thinking and Programming

制御構造の組み合わせ(復習)

Logical Thinking

- 第2回の授業で、制御構造について整理した
 - 複雑な処理を記述する場合は3つの制御構造を入れ子状に組み合わせる
 - どの制御構造も処理の入りが1つで、出口も1つなので可能

命令Aを入り口1つ、出口1つの箱として、入り口1つ、出口1つの分岐構造に置き換えることが可能

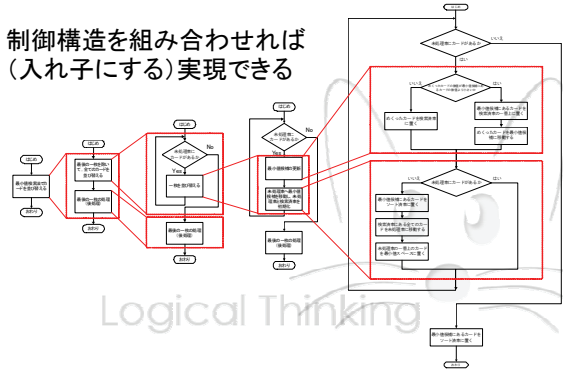


Logical Thinking and Programming

最小値選択法の例

Logical Thinking

制御構造を組み合わせれば(入れ子にする)実現できる



Logical Thinking

Logical Thinking and Programming