プログラミング演習 第 13 回演習レポート

2064229 福場脩真

# 指定された課題番号と課題名

第12回および第13回の基本課題

# (2) フローチャートあるいは自然言語によるアルゴリズムの記述

1.第12回基本課題1.「全探索」

テキスト

中程度の精度で自動的に生成された説明

2.第12回基本課題2.「線形探索」

ダイアグラム

自動的に生成された説明

3.第13回基本課題1.「二分探索」

ダイアグラム

自動的に生成された説明

# （3）アルゴリズムが「正しいこと」である説明あるいは証明

ずべてのプログラムについて、データの数以下の整数を入力すると、正しい最小値を出力した。それ以外の整数を入力すると、エラーメッセージを正しく出力した。よって、これらのプログラムは正しい。

# (4) ソース・プログラムの説明

1.第12回基本課題1.「全探索」

リストの指定した範囲の合計を計算するmysum関数を定義した。

メインの処理main関数について。データ処理を簡単化するため線形リストLinkedListクラスを作り、そのインスタンスとn、dataの合計total、最大値を格納するmax\_dを定義。リストからn人の組み合わせを作るitertools.combinationsメソッドでループを回す。最小値をtotalからx[0]の値からx[1]までの範囲でdataの数を合計した値を引いた値で設定する。

Xの長さ-1の範囲でループを回し、中で最小値を更新する。こうして得られた最小値と、その時の組み合わせを出力する。

2.第12回基本課題2.「線形探索」

組み合わせが存在するかを判別する関数puttableを定義。これはinterval、n、dataの3つを引数に取り、intervalはdataの合計totalから1までの範囲の整数をとる。Intervalとnについて、距離intervalで人数配置できるかを判別し、可能ならTrue、不可能ならFalseを返すようになっている。

線形探索をするMain関数では、nが不正でないかを判別したのち、dataの合計totalをループ処理する。ループで回し、puttableで判別することで線形探索になっている。

3.第13回基本課題1.「二分探索」

組み合わせが存在するかを判別する関数puttableは線形探索のものと同じ。Main関数で二分探索をする。Nが不正でないかを判別したのち、totalとlを初期値0、hを初期値totalで定義。L + 1がhより小さい間でwhile処理を回し、m をlとhの平均に設定。Mを引数としてputtableをじっこうし、Trueを返したらlにmを代入。それ以外ならhにmを代入する。こうすることで、lを開始点、hを終点として、範囲を絞りつつ二分探索ができる。

# (5) 考察

1．

全探索と線形探索を人数8人、データの数30人で実行し処理時間を比較すると、全探索は3分かかったのに対し、線形探索は1秒弱で完了した。全探索の場合、調べるデータが莫大でになるが、線形探索は該当するものが見つかった時点で処理を終了するので、これだけ処理時間に差が出たと思われる。

2.

線形探索と二分探索の処理時間を比較する。データ数150のd3と、同じデータ数で塔間距離が大きいd4について、それぞれ人数10人、100人で調べた。結果、d3の場合は線形探索と二分探索で処理時間はほとんど変わらなかったが、d4では線形探索は10人で43秒、100人で　6秒かかった。一方で二分探索の場合は1秒ほどで完了した。これは、2分探索は調べる範囲を半分に絞りながら探索するため、線形探索よりも解に辿り着くまで効率的に探索できるからだと考えられる。その差異は、計算量が多くなる10人の場合でより顕著に現れたと思われる。