

プログラミング技法

第6回演習

資料は

http://www-hiraki.is.s.u-tokyo.ac.jp/lectures/prog_giho/

プログラム高速化のまとめ

今日の課題

- 行列積のプログラム、またはバブルソートではないソートのプログラムを出来るだけ高速化せよ
 - 高速化前のプログラムに(コンパイラの最適化オプション以外の)高速化を施し、実行時間差を表にまとめる。
 - 注意: 高速化しやすいように最初のプログラムに駄目プログラムを使わないように

レポート課題のまとめ

課題1: スタックの総和を求める

課題2: 行列積を求める

課題3: バブルソートより速いソートを実現

課題4: 行列積またはソートの高速化

提出方法: 学籍番号、氏名を付けて

`proggiho-report@hiraki.is.s.u-tokyo.ac.jp`

まで送付

締め切りは全てのレポートについて学期末まで。

課題1

- 十分に大きい配列（例えば1000万要素）を取りスタックとして使う。つまり、操作する場所をスタックポインタとしてもつ。
- まず、配列にランダムまたは連続数で順次書き込む（プッシュ操作）。
- 次に配列を一番最近書き込んだものから順次読み出し総和を得る。
- 配列サイズを16から2倍ずつ大きくしたときの要素あたりの実行時間変化をリストまたはグラフとして求めよ。

課題2

- 十分に大きい2個の行列の行列積を求め、別の行列として作成するプログラムを作成する。
- 行列サイズを順次大きくし、演算あたりの実行時間変化をリストまたはグラフとして求めよ。

課題3

- アルゴリズムを変えて、バブルソートより速くする。
- バブルソートは計算回数が $O(n^2)$ の遅いアルゴリズム
- 良く知られている速いアルゴリズムは
 - Quick Sort
 - Merge Sort
 - Radix Sort などいろいろある。
- 本やWebページで探して、実装して速度比較

課題4

- 行列積のプログラム、またはバブルソートではないソートのプログラムを出来るだけ高速化せよ
 - 高速化前のプログラムに(コンパイラの最適化オプション以外の)高速化を施し、実行時間差を表にまとめる。
 - 注意: 高速化しやすいように最初のプログラムに駄目プログラムを使わないように