

レポート、コンテスト 提出の注意

レポート課題、コンテスト課題提出方法

- **日時:**
2020年2月3日(月)24時厳守
- **形式:**
PDF形式(レポート本体はPDFしか認めない)
 - コンテストのソースコードはZIPまたはtar+gz形式、コンテストの報告書本体はPDF
 - **名前、所属、学籍番号の記載を忘れずに!**
(時々書かない人がいる)
- **提出方法:** 電子メール: hanawa_AT_cc.u-tokyo.ac.jp
(_AT_=@)
 - **受理メールが届かない人は催促してください。**
そうしないと、未提出になるかもしれません。

レポート課題採点指標

- **基本点 (+20点)**
 - 以下の1~6がそろっている場合、基本点を加算する
 - 1.表紙(名前、所属、学籍番号)、2.問題の説明(問題のレベル Lxx を明記すること)、3.結果(実行時間に関する表、および、場合によっては1PEから1088PEまでの実行結果と台数効果の図)、4.結果の考察、5.講義を受けた感想、6.付録としてプログラムの主要部分
- **課題点(+1 ~)**
 - 問題のレベルを点として加算(妥当に回答した場合のみ)
 - 講義で解説したプログラムの場合、そのまま載せただけでは加点しない
 - 基本問題と上級問題を両方答えている場合には、上級問題分しか加点しない
 - あとは常識で判断してください
- **独創点(+5 ~)**
 - 図が奇麗、説明がうまい、説明が丁寧、いっぱい実験(実装)している、よくできた考察、など
- **出席点(~+20)**
 - 切り上げ(出席日数/12) × 20 点
- **遅刻点(-10、-30、-∞)**
 - 1日目:-10点、2日目:-30点、3日後以降は-∞点。
- **特に、「講義の感想」を忘れないこと!**

コンテスト課題採点指標

- 提出物

1. コンテスト課題出力結果
2. 解法、実装法に関する説明(1ページ以上)
3. ソースコードをtarで固めたもの
 - 1と2は、PDFの形式で

- 入賞点(+100)

- 1位~3位に入賞した人

- 完走点(+20)

- 出力結果が正しいプログラムを作成した場合のみ加点
- Fortran版、C版、別に順位を付ける
- Fortran版、C版、双方に出場できる(双方加点あり)

総合評価基準

- <レポート課題点>と<コンテスト課題点>の合計で、総合評価する
- 0 ~ 39
 - 不可
- 40 ~ 59
 - 可
- 60 ~ 79
 - 良
- 80 ~
 - 優

コンテスト課題発表

その前に

- コンテスト参加者のメーリングリストを作るため、参加予告メールを送ってください。
 - **Subject: SPC2019aAppl** と書くこと。
 - 名前、学籍番号、メールアドレス を書くこと。
 - 宛先：
hanawa_AT_cc.u-tokyo.ac.jp (_AT_=@)
 - メーリングリスト名：
spc2019a@cspp.cc.u-tokyo.ac.jp
 - 本メーリングリストで、質問などを受け付けます。
 - 参加者間の情報交換にも使ってOKです。

問題説明

- 課題:「複数の右辺 **b 」がある「LU分解」**

- 連立一次方程式

$$A x = b$$

の解ベクトル x を求める

- ここで、解ベクトル x が1本である保証はない
- すなわち、 m 本の解ベクトルをまとめた行列 X を

$$X = (x_1 \ x_2 \ \dots \ x_m)$$

とし、 m 本の右辺ベクトル b をまとめた行列 B を

$$B = (b_1 \ b_2 \ \dots \ b_m)$$

とすると、

$$A X = B$$

の解ベクトル行列 X を、解く問題と定義する。

コンテストプログラムの実行

コンテストプログラムの実行 (C言語/Fortran言語共通)

- 以下のコマンドを実行する

```
$ cp /work/gt34/z30105/spc2019a.tar.gz ./
```

```
$ tar xvfz spc2019a.tar.gz
```

```
$ cd SPC2019a
```

- 以下のどちらかを実行

```
$ cd C : C言語の人
```

```
$ cd F : Fortranの人
```

コンテストプログラムの実行 (C言語/Fortran言語共通)

```
$ cp spcsamp.h spc.h
```

```
$ make
```

- ```
$ pjsub spc.bash
```
- 実行が終了したら、以下を実行する

```
$ cat spc.out
```

# コンテストプログラムの実行(C言語)

- 以下のような結果が見えれば成功

-----  
N = 1088 , M = 1088

LU solve time = 6.124222 [sec.]

141.133095 [MFLOPS]

Pass value: 1.763916e-02

Calculated value: 3.462407e-06

OK! Test is passed.  
-----

# コンテストプログラムの実行 (Fortran言語)

- 以下のような結果が見えれば成功

-----  
NN = 1088

MM = 1088

LU solve time[sec.] = 9.14490485191345

MFLOPS = 94.5149703209670

Pass value: 1.763916015625000E-002

Calculated value: 3.462407026655001E-006

OK! Test is passed.  
-----

# サンプルプログラムの説明

- `spcsamp.h` の中身
- `#define N 1088`
  - 数字を変更すると、行列サイズが変更できます
- `#define M 1088`
  - 数字を変更すると、右辺ベクトルbの本数が変更できます

# コンテスト課題提出方法

## 【コンテスト課題実行方法】

1. `spc.c / spc.f90` の中の関数(手続き) `spc` を並列化してください。
2. `spcFINAL.bash` を実行してください。
  - `pjsub spcFINAL.bash`

## 【提出物】

1. 実行後作成される、`spcFINAL1.out`、`spcFINAL2.out`、`spcFINAL3.out`、`spcFINAL4.out`が、提出用の出力リストです。
2. さらに、ソースコードの提出が必要です。

# コンテスト課題プログラムの実行方法 (C言語/Fortran言語共通)

```
$ pjsub spcFINAL.bash
```

- 実行が終了したら、以下を確認する

```
$ cat spcFINAL1.out
```

```
$ cat spcFINAL2.out
```

```
$ cat spcFINAL3.out
```

```
$ cat spcFINAL4.out
```



# コンテスト課題採点方法

- `spcFINAL1.out`～`spcFINAL4.out` の実行すべてが、エラーなく実行されれば、**予選通過**です。
- 予選通過者に対して、`spcFINAL1.h`、および `spcFINAL2.h` で定義された問題の各実行時間について、高速なものから以下の配点をします。
  - 1位: 10点
  - 2位: 5点
  - 3位: 2点
  - 4位: 1点
  - 5位以下: 0点
- 配点の大きい順にソートし、上位1位～3位が入賞です。

# コンテストの注意点

1. `spc.c` / `spc.f90` 中の関数(手続き)  
`spc` 内のみコードの変更が可能です。
2. メイン関数内は、変更不可です。
3. 計測ルーチンをいじってはいけません。
4. `spcFINAL.bash` を原則変更してはいけません。  
(ただし、ハイブリッド実行のための記述と、グループ名、キュー名の変更は可能です)

# コンテストの注意点

5. 解ベクトルが1-ベクトルとなるように右辺bを生成しています。したがって、**解ベクトル行列x[][]に1を代入して終了**、というような、**<解ベクトルを知っている実装>**をしてはいけません。
  - 数値解法は、何を使っても構いません。
  - ただし、反復解法を利用する場合、初期ベクトルは0-1の一様乱数で生成してください。
6. その他、不正と判断された場合は、失格とします。
7. コンテストの注意点の内容について不明な点は、メーリングリストで公開質問してください。

# その他の注意点

- C言語版では、右辺Bの収納配列は、**行列Bを転置した形式**で収納されています。
  - C言語では列方向アクセスが非連続アクセスになる。そのため、Fortran言語の実装に対して、性能劣化する原因となるため。

# コンテストの注意点

- 以下を除いては、何をしてもOKです。
    - LU分解以外の数値解法の実装
    - コンパイラオプションの変更
    - 数値計算ライブラリの利用
    - アセンブラの利用
- など