2004 年度情報処理 II　第 12 回

まとめ

桂田 祐史 (かつらだ まさし)

2004 年 7 月 15 日

「情報処理 II トップページ」[1]

1　連絡事項

- この講義のレポートは遅れても、提出することにはそれなりの意義がある、という考え方で採点評価していますが、これ以上遅れては意味がないという dead line があって、それは 7 月 23 日 (金曜) です。
- 時々レポート・メイルのリストが化けます。その原因は送られたメッセージの中に ISO-2022JP でない文字 (ローマ数字の 'II' や半角の '「' や '…') が含まれているからです[2]。本当はそういう文字を送らないようにメイラーが出来ているべきですが、こちらもそれが来ても大丈夫なようにしかけを作っておくべきだったかもしれません。
- 人の答を写すのか、独特の間違いが複数あるのは…しかもそういうのって、到着順が並んでいたりして… 人に尋ねるにしても正しい答を選んでコピーするのも能力のうちか…

  人に教えてもらって構わないけれど、自分で理解して、自分で料理すること

- 一つの課題でいくつもレポートを送ってくる人がいましたが、原則として一つであるべきです (もちろん、レポートを訂正するためのメッセージは問題ありません)。採点する側の便宜を考えましょう。

2　課題 0

これは名簿作りのアンケートでした。ご苦労様。

---

[1] http://www.math.meiji.ac.jp/~mk/syori2/

[2] ローマ数字はローマ字を使って (例えば II は I を二つ並べる) 書くべきです。「や…は最初変換するときに注意すれば後は間違えないでしょう。もっとも、この種のことを人間が注意するのにも限界があるので、本来はソフトウェアの側で対応すべきことだと私は考えます。

3 課題 1

友人との間で、電子メイルでメッセージのやり取りをする。その際、「返信」をして、お互いのメッセージを引用しあって、「会話」する (大した内容は必要ない)。複数回メイルのやり取りをした結果を `syori2@math.meiji.ac.jp` まで、電子メイルで送信する。

第三者 (桂田) に見せるわけで、内容には注意すること。普通は相手の了解を取る必要がある (この授業中に隣の友人とやり取りする場合は必要ないであろう)。

数年前と異なり GraceMail を使っていることもあって、問題なくこなしている人が多かったです。

4 課題 2

次の質問に答えよ。

(1) 明治大学のドメイン名は？

(2) 株式会社 SONY のドメイン名は？

(3) 明治大学理工学部数学科サブドメインのドメイン名は？

(4) 自分の情報科学センターのアカウントの E-mail アドレスは？

(5) MIND アクセス・レベル 3 のホストを一つあげなさい。

(6) (これまでに検索エンジンを使ったことがあって、差し支えがなければ) どの検索エンジンを使っているか、その理由は何か、主にどういう用途に使っているか、教えて下さい。

(1) は `meiji.ac.jp`, (2) は `sony.co.jp`, (3) は `math.meiji.ac.jp`, (4) は 自分のログイン名に `isc.meiji.ac.jp` をつなげたもの。例えばユーザー名が `ee380`xy ならば、`ee380`xy`@isc.meiji.ac.jp`

(5) WWW サーバーのように学外からアクセスできるホスト、例えばインターネット講習会で説明される sagami2.isc.meiji.ac.jp や WWW サーバーになっている www.meiji.ac.jp など。あるいは、プロキシー・サーバーもそうである。例えば `ikuta-p.mind.meiji.ac.jp` など。

(5) で oyabun をあげた人が多いけれど、oyabun はレベル 3 ではありません。また www と書いた人もいましたが、www はホスト名でないので、`www.meiji.ac.jp` (これはいわゆる別名ですが、この名前に対応するホストは一意的に決まります) と書かないと間違いです。

5 課題 3

自分の名前を構成する各々の文字 (桂田祐史なら「桂」,「田」,「祐」,「史」の 4 文字) の JIS コード, EUC コード, SJIS コードを調べよ。

実はすごく出来が悪くて驚いています。例えば、

	EUC	JIS	SJIS
桂	b7 cb	37 4b	8c 6a
田	c5 c4	45 44	93 63
祐	cd b4	4d 34	97 53
史	bb cb	3b 4b	8e 6a

のような回答を期待していました。

`od -cx` の実行結果だけ示した人がいますが、これは問題に答えたことになっていません。何か実験なり測定なり調査をする必要があった場合に、結果を出すだけではだめです。「見れば明らか」とは限りません。

問題文で、「各々の文字」と言ってあるので、一つ一つの字についてコードを答えないといけません。口頭でも何度か注意したことです (授業中に内職をする習慣なのかもしれませんが、「課題について説明します」と言っているときくらいは真剣に聴きましょう)。

mydump2.c[3] という C プログラムをコンパイル&リンクして、

```
isc-xas06% gcc -o mydump2 mydump2
isc-xas06% cat kanji.txt | ./mydump2
```

とすると表示されます。

```
mathweb% ./mydump2 kanji.txt
        EUC     JIS     SJIS
桂:     b7 cb   37 4b   8c 6a
田:     c5 c4   45 44   93 63
 :      20
祐:     cd b4   4d 34   97 53
史:     bb cb   3b 4b   8e 6a
C-J:    0a
mathweb%
```

6 課題 4

文中のアルファベットの出現頻度は 'e' が一番高く、その次は… などと言われ、古典的な推理小説の暗号の話の種になったりしている。Gutenberg Project の中のテキストで、そのことを確かめて見よ。手作業ではなく、なるべくコンピューターにやらせること。テキストごとに大きな違いがあるか？文字が別の記号に置き換えられた場合、出現頻度情報から解読することの可能性について論ぜよ (要するに他の文字の出現頻度はどの程度まで一定しているのか調べる – 実際に試してみると良いのだけど)。なお、文字の頻度を調べる `hindo.c` というプログラムを用意した[4]。(このプログラムは文字の出現頻度順には表示しないが、`sort` を使えば簡単に頻度

[3] `http://www.math.meiji.ac.jp/~mk/syori2/mydump2.c`
[4] `hindo.c` は `filter` ディレクトリィに入っている。

順に並べられる。どうすればいいか？今回説明した話の簡単な応用である。)

hindo.c のコンパイルと使用例

```
isc-xas06% gcc -o hindo hindo.c
isc-xas06% cat hindo.c | ./hindo
```

テキスト、あるいは作家ごとに単語の使用頻度の癖のようなものがあると思われるが、そのことを Gutenberg テキストで実際に調べてみよ。ルイス・キャロルとマークトウェインの書いたものにどの程度の差があるか？

前半 (文字の出現頻度) については、複数のテキストについて、出現頻度の上位 5 つくらいまでを並べた表を作って、それを参照しつつ論じるというのが一つの解答ルートでしょう。

```
80day10.txt   etaon
aesop11.txt   etaoh
alad10.txt    etaio
alice29.txt   etaoh
anne11.txt    etaon
frank11a.txt  etaon
hfinn10.txt   etoan
moon10a.txt   etoai
sawy210.txt   etaon
sawy311.txt   etaon
sawyr10.txt   etoan
wizoz10.txt   etoah
```

のような表を作れば (ちなみにこの表は半自動的に作りました)、何か言えそうだと分かるでしょう。

filter ディレクトリィにある 12 個の小説のテキスト・ファイルでは、いずれも e の出現頻度が 1 位, t の出現頻度は 2 位. 3 位は大抵 a であり、テキストによっては o であるが、その場合も a は 4 位に入っている、くらいは言えそうです。

全部混ぜて測ってみるとどうだろう？と思ったら

```
oyabun% cat *.txt | ./hindo | sort -n -r +1 | head -5
e: 362970 (12.2%)
t: 268142 ( 9.0%)
a: 233099 ( 7.8%)
o: 225494 ( 7.6%)
n: 203841 ( 6.8%)
```

これを見ると、a と o はコンマ以下の争いで逆転が起こることもうなずけます。

出現頻度の平均や分散を調べるのも良いかもしれません。

単語についても同じような調子で調べてみるわけです (省略します)。

復号 (解読) するスクリプト decode.sh[5] を置いておきます[6]。
ちなみに

```
isc-xas06% ./hindo angou.txt | sort -n +1 -r|head -5
u:  1789 (12.8%)
o:  1197 ( 8.6%)
g:  1150 ( 8.2%)
m:   990 ( 7.1%)
n:   939 ( 6.7%)
```

の結果から u, o, g, m が実は e, t, a, o ではとあたりをつけて

```
cat angou.txt | /usr/ucb/tr uogm etao
```

とかするわけです。それからこれは “the” だろうと当りをつけて、n を h に置換して…という調子でやっていきます。単語の頻度を見るのも良いかも

```
isc-xas06% cat angou.txt | /usr/ucb/tr uogmn etaoh | ./top20x
 281 the
 132 awr
  88 he
  83 oy
  73 a
  71 to
  65 V
  58 dv
  55 dw
(略)
isc-xas06%
```

“awr” は “and” かな？ “oi” は “on” かな？だとすると w と n に…そろそろ色々な単語らしきものが見えてきます。それでまた当りをつけて…(以下略)

7 課題 5

以下のことを調べよ。〆切は 7 月 2 日 (まだ本決まりでない) とする。(ファイルのサイズについての感覚を身につけてもらうのが主旨であって、自分で計算すること。)

(1) フロッピー・ディスクを使ったことがあるか？(教えて下さい。アンケートのつもりです。)

(2) 自分が触れるコンピューター (情報科学センターの Windows 環境, UNIX (Solaris) 環境,

[5] http://www.math.meiji.ac.jp/~mk/syori2/decode.sh
[6] まったくの余談ですが、高校生の時に英語の授業でこのテキストの暗唱をさせられました (長いんで「うげー」と思いましたが、今となってはちょっと懐かしい)。

自宅のパソコン) にあるファイル[7]のサイズについて調べよ。バイト数以外に、CD-R にどれくらい入るかを記せ。

(a) 文書ファイル
レポート、メイル、C プログラム、TEX のソース (.tex) など。
(単にサイズだけ書いてもあまり意味がない。長いものもあれば短いのもあるのだから。例えば「印刷して何ページくらいの文書が何バイトになる」等の情報を添えること。)
ワープロソフトの文書ファイルなどを調べてみるのも良い。

(b) 実行可能プログラム

- 自分が普段使っているプログラムをいくつか選び、そのプログラム・ファイル[8]のサイズを調べよ。(大規模なソフトウェアの場合、一つのプログラムから別のプログラムを呼び出し、全体として複数のプログラムが協調して働くこともあるので、結構難しい。それゆえ必修とはしないが、トライしてみること。)
- C で書いたプログラムを持っている場合、コンパイル前 (ソースプログラム) と後 (実行可能プログラム、あるいは機械語プログラム) でどうサイズが変わるか。

(c) 現在、自分が持っているファイルの総量。それは自分のホームディレクトリィのあるディスクの全容量の何 % に相当するか。

UNIX 環境で調べる場合

```
isc-xas06% du -ks ~
isc-xas06% du -ks ~/.snapshot
isc-xas06% df -k
```

Windows 環境から調べても良い (どうやれば良いかは自分で見つける)。

(3) 自分が持っている本を一冊選び、その文字情報を記憶するファイルを作った場合、サイズはどれくらいになるか計算せよ。CD-R には何冊分記憶できるか。(古い小説などの場合、実際に青空文庫で電子化されたファイルが探し出せるかもしれない。自分の計算と照らし合わせると面白い。)

(4) (もし出来れば) 画像ファイル、音声ファイルなど。これはパソコンに限らない。
(記録の形式、画像の場合は図の大きさ (ピクセル数) & 色数、音声の場合は時間等も分かる範囲で調べる。本来のサイズの何分の一に圧縮されているか概算せよ。)
カメラつき携帯を持っている人からのレポート求む (私に色々教えて下さい)。

この課題のレポートは遅れがちですね。ですから、注意事項を。
(1) はフロッピーディスクは遠くなりにけり、ですね (まだコンビニには置いてありますが)。
(2) (a) について。ワープロの文書ファイルを調べた人が多かったです。どういうファイルを調べたのか書いていない人が結構いました (いきなり「○○バイトのファイルがあった」だ

[7]自分以外の持ち物でも構わない。例えば、私はホームディレクトリィ (~re00018) を開放している (読み出しを許可してある) ので、そこにある Gutenberg テキストや、解析概論 I の講義ノートの LATEX ファイル (~re00018/tex-sample/textbook/ にある) などを調べることも出来る。

[8]プログラムの実体は、UNIX ならば which プログラム名 として追跡する。Windows ならばアイコンを右クリックしてプロパティから追跡する。

け)。単にバイト数を書いても意味がありません。文章だったら、何文字 (正確に数えなくても良いです) の記録に何バイトくらい必要か、それは記録メディアにどれくらい入るものか、そういうことを各自が理解できて欲しいのです。と最初に小言ばかり書きましたが、ワープロ (Word) など、こちらが知らないことを教えてもらって「ほほー」と思わせてくれた回答も結構ありました。

(2) (b) について。これはまじめにやると結構難しいです。大ざっぱに言って、ソースプログラムのサイズ x と、実行可能プログラムのサイズ y は一次式の関係 $y = ax + b$ にあると想像できますが、a はどれくらいか知って、自分が日頃使っているプログラムが、どんなに大きいか分かってもらえたら、と思って出題しました。なお、x と y は単純な比例関係にはありません。例えば C で何もしないプログラムを作ってもサイズは 0 ではありません (結構大きいです)。だから「コンパイルすると大きくなった」は確かにそうなったのでしょうが、いつでもそうなるとは限らないことです。…これについては、こちらの準備が間に合いませんでした。

(2) (c) について。こちらから学生のファイルについて確認できないのですが、明らかに「おかしい」回答が結構あります。各ユーザーが持てるファイルの量には制限があって、今年度のセンターの場合 50MB のはずです。この制限を数百倍破っている人がいましたが、これは単位を間違えたのでしょう。こういう勘違いをしない人になって欲しいのですが。

(4) 携帯の音声ファイルというのがありました。これは気がつかなかった (使っていないもので)。ちなみにこのたび、デジカメを買ったので、僕自身がレポートを書いてみようと思います。いつか情報処理 II のページを再訪することがあれば探して見て下さい。

8 課題 6

引用すると長くなるので、URL のみ示します。`http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2004-09/node22.html`

Newton 法や二分法で方程式を解くという課題ですが、まだあまりレポートが届いていません。ですから、この課題についての解説は後日出します。すべての課題に関するレポートを出さなくても単位は出ますが、なるべく解いて提出してください。

課題 6 以降のレポートの締め切りは 7 月 23 日 (金曜) まで延長します。

9 課題 7

パソコンで Mathematica が使えるようになって、実習がぐっとしやすくなりました。もう少し準備しておけば、突っ込んだことが出来たかな、と反省しています。一度体験しておくと、後でハードルが低くなると思うので、ぜひ実際に自分で試して、レポートを提出してください。