

Teach Yourself KMC

独習KMC vol.2

京大マイコンクラブ 著



巻頭言

須田 淳

今を遡ること 22 年、平成元年に KMC (京大マイコンクラブ) 第 12 代会長を務めさせて頂いた須田淳と申します。京都大学工学部の電気電子工学科 (通称、電電) で准教授*1をしています。KMC 顧問兼現役部員最長記録保持者の上原哲太郎先生 (Tetsu=TaLow さん) が霞ヶ関の某省へ異動となりましたので、今年から顧問の任を引き継ぐことになりました。今回、『独習 KMC vol.2』の発行に際して巻頭言の依頼がありましたので、私自身の KMC への思い、KMC を卒業して 20 年経って感じていることを書きたいと思います。

この冊子を見れば一目瞭然かと思いますが、KMC は様々な才能や技術、技能を持ったクリエイター達の集団です。凄腕のプログラマー、ハードの達人、ゲームデザイナー、作曲・アレンジャー、演奏家、イラストレーター、漫画家、看板職人、文筆家、編集者、交渉人……。本当に凄い奴らです。入部時点で既に凄いレベルの奴もいますし、入部して他の部員と交流する中で技術を身につけ、覚醒する奴もいます。興味の対象が四方八方を向いておりますが、そんな彼らを束ねているのは、コンピュータを愛するという共通項でしょう。私も現役時代は、アニメーションを作ったり、イラストやマンガを『KMC News』(年数回発行されていた会報) に描いたり、ワンボードマイコンを作製したり、占いソフトを開発したりと存分に活動しました。技術を互いに競い合ったり、互いに教え合ったりという切磋琢磨、相互作用が当時の KMC にありました。今でもその伝統はしっかり息づいているようです。昨今、最先端研究において異分野融合という言葉がもてはやされていますが、20 年前の KMC はそれを先取りしていたと思います。

KMC の創立から 35 年が経過し、たくさんの OB/OG が社会で活躍しています。企業において新しい製品やサービス、事業を作り出すことも、大学や研究所で新技術の研究開発を行うことも、その原動力は、新しいものを生み出したいという情熱です。若い頃、怪しいこと(?)に燃やした情熱が、今や、社会を支える・前進させる仕事に振り向けられているのです! この冊子に記事を書いているみなさんの 10 年、20 年後が本当に楽しみです。

ところで、最近 Facebook に KMC のグループができました。十数年の時を経て、KMC で同じ

*1 専門は半導体です。京都大学 OCW で担当する半導体工学の授業の初回が公開されていますので興味のある人はご覧下さい (元電子工作少年のプライドをかけた授業を目指しています。まだまだ修行中ですが)。 <http://www.youtube.com/watch?v=yDsUB75vydE>

時期を過ごした仲間達の日常が（人によってはその日のお弁当のおかずまで、まさに手に取るように）分かるようになりました。驚くべきことは、怪しかったあの面々が（自分もその一人かも知れませんが）、皆、良いパパさんになっているということです。DNA にハードコーディングされた子育てプログラムが発動したのでしょうか、それとも、神の御業なのでしょうか。いろいろ考えましたが、大いなる時の流れのおかげだろうとの結論に至りました。三十路を過ぎて怪しさを維持できる人はあまりいません。年齢と共に、怪しさが薄れれば、優れた技を持ったクリエイターである KMC OB が良いお父さんになるのはむしろ必然と言えるでしょう。2 つほど例を挙げてみましょう。

「絵が得意」

絵が上手なお父さんは、小さい子供にとって自慢のパパです。私自身の経験ですが、娘が保育園に通っていた頃は、迎えに行くたびに、娘とその友達に取り囲まれプリキュアを描かされました。娘は「うちのパパって凄いでしょ」と得意顔です。ここで注意ですが、（画風にもよりますが）真剣に書きすぎると NG です。おジャ魔女ドレミなら大丈夫ですが、プリキュアは（バージョンによっては）注意が必要です。他のお母さんや保育士さんから、「お父さん、絵はすごく上手だけど……。」と違う意味で一目置かれてしまいます。コミカルなタッチでさっさと書きましょう。これだと、周りの方からも、「あら、お父さんマンガがお上手ですね」と素直に感心してもらえます。

「遊びも真剣」

年と共に怪しさは薄れても KMC 魂は薄れることはありません。何か探求する題材があると子供のように熱中します。LEGO やプラレール、夏休みの自由研究や自由工作など、子供と一緒に熱中します。子供のためという以上に、自分でもいろいろやりたいので、道具をそろえることに散財も惜しみません。子供にとっては、太っ腹のスポンサー兼ガチで真剣に遊んでくれる最高の遊び相手となります。子供の創造力を大いに引き出すことでしよう。と言うことで、これを読んでいる女性のみなさん（いるのでしょうか？）KMC 男子部員は狙い目ですよ！ KMC 女子部員はどうかって？ うーん……。すぐ隣に 1 名いますが、ノーコメントとしておきましょう。

話が脱線しましたが、この独習 KMC は、とにもかくにも凄い奴らが作った素晴らしい本です。是非、お買い求めいただければ幸いです。今後とも、皆さま、どうか KMC をお引き立ての程、よろしく願います。

目次

巻頭言 (須田 淳)	i
とびうおのリア充物語——序文 (@tobiuo12730)	1
KMC の最近の活動	3
後期勉強会紹介 (勉強会参加者一同)	4
コンピュータ・ジオメトリ読書会	4
TAPL 読書会	5
Lisp 勉強会 (PCL/PAIP 読書会)	5
SICP 読書会	6
JavaScript で人権を作ろう 2011	6
イラスト品評会	7
とうけい!	7
Rails3 で Web サービス	8
DTM 練習会 2011	9
とびうおのリア充物語——沖縄編 1 (@tobiuo12730)	10
特集: KMC 部員と Twitter クライアント	13
KMC 部員の Twitter クライアント (@c-possum)	14
とびうおのリア充物語——沖縄編 2 (@tobiuo12730)	16
部員のコラム	21
Haskell Golf 攻略 (@henkma)	22
はじめに	22
Anarchy Golf の紹介	22

Golf と Haskell	23
Haskell Golf の基本	24
基本戦略	25
リスト	27
思考方法・方針紹介	29
おわりに	34
【禁書 SS】インデック Scheme (@l_possum)	35
不動点の話 (@t_uda)	40
ひでシスの多様性物語——インドネシア編 (@hidesys)	45
とびうおのリア充物語——沖縄編 3 (@tobiuo12730)	50
あとがき	53
とびうおのリア充物語——沖縄編 4 (@tobiuo12730)	54
編集後記 (@d_possum)	55

とびうおのリア充物語——序文

@tobiuo12730

序文

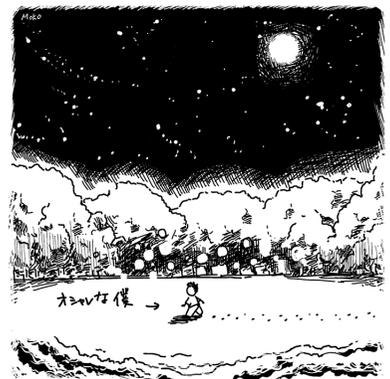
長期であれ短期であれ、壮大であれ些末であれ、人間は何か目標を持ち、それについて計画を練り、計画に基づいて行動するのが理想だと思います。僕の場合たまたま目的はリア充になることでした。それを実現するための計画は

- (1) ユニクロに入店してオシャレになる
- (2) 合コンに誘われる
- (3) 彼女ができる
- (4) リア充になる

というごくシンプルなものでした。今でもこの基本方針は変わっていませんが、計画発案当時の若かりし僕には (1) と (2) の間に様々な困難があることを知りませんでした。

今年の春先にそれまで恐れていたユニクロに入店し文明人とは思えない服装を卒業した後、僕には自分の生き方について考えざるを得ないような大きな出来事が何回かありました。今回はそれらの出来事のひとつ、沖縄で恋に落ちた女の子の家にホームステイしに行き、告白する前に玉砕したお話をします。白いサンゴ礁が広がり、ヤンバルの自然林が広がる小さな村、東村の小中学生にかなり知れ渡ったこのお話をどうぞお楽しみください。あと誰か合コンに誘ってください。

(このコラムは各章の終わりに1章ずつ掲載されます。部誌をお読みの際に息抜きとしてお楽しみください。)



KMC の最近の活動

後期勉強会紹介（勉強会参加者一同）	4
コンピュータ・ジオメトリ読書会	4
TAPL 読書会	5
Lisp 勉強会（PCL/PAIP 読書会）	5
SICP 読書会	6
JavaScript で人権を作ろう 2011	6
イラスト品評会	7
とうけい！	7
Rails3 で Web サービス	8
DTM 練習会 2011	9
とびうおのリア充物語——沖縄編 1（@tobiuo12730）	10

後期勉強会紹介

勉強会参加者一同

2011 年 10 月以降に KMC で活動している勉強会を紹介します。

コンピュータ・ジオメトリ読書会

理論寄りな本の読書会です。

アルゴリズム分野の 1 つとして「計算幾何学」というものがあります。これは幾何学っぽい問題を如何にコンピュータで効率よく扱うかということに焦点を当てた分野です。『コンピュータ・ジオメトリ』^{*1}は計算幾何学を扱う本の代表的なものの 1 つであります。

具体的にどういう問題があるかを挙げてみます。

線分が 100 万本くらい与えられてその交点を全部列挙するというのを考えます。交点の数は「そんなに多くない」と保証されているとします。このとき、愚直なアルゴリズムでは「すべての異なる線分のペアについて交点を持っているか判定し、交点を持っているならそれを報告する」というものが考えられます。しかしこれでは $100 \text{ 万} \times 100 \text{ 万}$ で相当な時間を取ってしまいます。やったことがないのでわかりませんが少なくとも数時間を要するのではないのでしょうか。一方で、この本に紹介されている「平面走査」という手法を使うと、これが $100 \text{ 万} \times \log(100 \text{ 万})$ で終わり、恐らく 2 秒程度で終わるものだろうと思われれます。

他にもあります。多角形の 3 角形分割・ボロノイ図・3 次元の凸包などです。これらはいずれも、普通に処理すると効率が悪くなってしまふものを如何に効率よく処理するかということに焦点が当てられています。近年ではコンピュータの扱うデータの量というのはどんどん大規模になってきていますし、大規模なものを如何に高速に扱うかについて知ることは今後それなりに役に立つものなのではないかと思っています。多分。

@ir5

^{*1}Mark De Berg 他、浅野哲夫訳、近代科学社、2010、ISBN 978-4-7649-0388-3

TAPL 読書会

近頃、OCaml や Coq といった関数型言語や定理証明系がなにかと話題に上るようになってきましたが、このような静的型付け関数型言語で用いられている型システムや論理の理論的背景を学ぶために、3-4 回生の 5 人程度で *Types and Programming Languages*^{*2} (TAPL) の読書会を行っています。

この本では、関数型プログラミング言語の基とするラムダ計算という計算のモデルを取り上げ、これを実用的なプログラミング言語へと拡張していく中で、型システムを与えることによってプログラムに対してどのような安全性を保証できるかということを議論しています。理論について説明する合間に実際に ML を用いて言語処理系を実装する章が挟まれているなど完全に理論寄りではなく、処理系の実装や現実世界のプログラミング言語についても触れていることから、プログラマの多い KMC でも読みやすいかと考え読書会の教科書として選びました。

今年の 5 月ごろから読み始めましたが参加者の都合がなかなか合わせられず、12 月初めの時点で第 12 章までしか進められていません。全 32 章を読み切るにはまだまだ時間がかかりそうです。

@k_hanazuki

Lisp 勉強会 (PCL/PAIP 読書会)

明冷の候、当記事をお読みの皆様におかれましては益々ご健勝の事とお喜び申し上げます。門ごとに飾られたしめ飾りが新しい年の訪れを言祝ぐ中、当 Lisp 勉強会の本年の締めくくりとして、その趣向をお知らせできますことは大変嬉しくあります。

さて、かねてより馬手に SBCL、弓手に PCL を持って見えざる多言語勢力を相手にちぎっては投げ、ちぎっては投げの奮闘を致しておりました『実践 CommonLisp』^{*3} 読書会ではございますが、目標としていたストリーミングサーバの作成を終了し、先般めでたく大団円を迎えることができました。つきましては主催者に代わり、日頃のご支援に心から感謝申し上げます。

またこれに引き続き、当勉強会では AI の分野を新たな地平と定め、参考書に『実用 Common Lisp——AI プログラミングのケーススタディ』^{*4} を採用して新たに読書会を開催することと相成りました。やはり少数者ではあるものの忌憚ない討論と質問が交され、この分野の代表的な(また記念碑的な)テーマについて学ぶとともに、より深い Lisp への理解を得られるものと確信しております。2012 年も当勉強会は肅々と活動して参ります。

yukiharu

^{*2}Benjamin C. Pierce. The MIT Press. 2002. ISBN 978-0-262-16209-8. <http://www.cis.upenn.edu/~bcpierce/tapl/>

^{*3}Peter Seibel, 佐野匡俊・水丸淳・園城雅之・金子祐介訳、オーム社、2008、ISBN 978-4-274-06721-1

^{*4}Peter Norvig, 杉本宣男訳、翔泳社、2010、ISBN 978-479-811890-1

SICP 読書会

計算機科学の古典的な教科書である *Structure and Interpretation of Computer Programs*^{*5} (SICP) の読書会です。

この本では、プログラミング言語 LISP の方言 Scheme が用いられ、アルゴリズムやデータ構造、抽象化、再帰、インタプリタ・コンパイラの実装など計算機科学の様々な概念について説明されています。プログラミングの世界ではとても有名な本で、著名なプログラマが必読書としてこの本を挙げることも多いです。

読書会は 1 回生から 4 回生まで幅広い参加者で行っていて、現在全 5 章のうちの 2 章を読み進めています。SICP 読書会は過去にも開催されたことがあるのですが、前回 (3 年前) の読書会では最後まで読み進めるのに 1 年以上かかったそうです。まだ始まったばかりなので読み終わるまでにはまだまだ時間がかかりそうですが、上回生の助けを借りながらゆっくり読み進めていきたいです。

@uiureo

JavaScript で人権を作ろう 2011

国連が、インターネットに接続することは人権の一つである、と宣言したそうです。僕たちインターネット利用者の多くはブラウザを通して Web を利用しています。Web の標準プログラミング言語としてどんなブラウザにも搭載されているのが JavaScript です。新しい技術が開発されるに従い、その重要性は高まってきており、今やそれなしで Web サイトを開発することは考えられません。Web 開発者にとって JavaScript を使えないということは、人権がないのと同じだと言っても過言ではないでしょう。

悲しいことに、僕も人権のない者の一人です。JavaScript を中途半端にしか使えないため、Web サービスを作ろうとしても拙い UI しか作ることができず、悲しい想いに暮れる夜も多いです。

このような悲しみを乗り越えるために、部内の人権のない者たちで身を寄せ合い JavaScript の勉強会を行っています。この勉強会の目的は、JavaScript を自由自在に操れるようになって、人権を獲得し幸福を実現することです。現在のところ、上回生に講師をしてもらい他の参加者が教わるという形で勉強会を進めています。まだ基礎的な文法を習っただけなので、人権獲得までの道は遠そうですが地道に頑張っていこうと思います。

JavaScript という人権を手に入れて立派な人間としての第一歩を踏み出す。そのことを僕たちは夢に見ています。

@uiureo

^{*5}Harold Abelson and Gerald Jay Sussman. The MIT Press. 1996. ISBN 978-0-262-01153-2. <http://mitpress.mit.edu/sicp/>

イラスト品評会

主な活動のひとつとしてゲーム制作を行っている KMC において、絵を描く人間は必要不可欠といっても過言ではありません。この勉強会は、絵描き育成の場・絵を描きたい人への機会提供の場として、年度初めから行っているものです。

勉強会は、宿題として与えられたテーマで絵を描いてきて、勉強会の場でその良し悪しを批評する、という形で進行します。批評される側が改善点を発見して能力を高めるのは勿論のこと、批評する側にも絵の良し悪しを見極める能力、適切な批判（基本を習得できていない者に応用的技能を教えても効果が薄い）をする能力を育むことも、この勉強会の目的です。

勉強会の様子はというと、メンバーひとりひとりのイラストがスクリーンに映され、10人弱、学年の差のあるメンバーが遠慮なく批評します。議論が白熱しすぎてポッコポコにされることもしばしばという具合です。それでも誰一人として心折れることなく絵を描いてきてくれます。まさに修行です。毎週講義のある水曜日に活動しているため勉学が忙しい者は満足に絵を仕上げられなかった、ということもしばしばですが、単位を投げ捨てながら時間を賢く使いながら画力を磨いております。

今年は絵を描きたいという新生が比較的多く、その需要に見合った勉強会を開催できてよかったです。

crys

とうけい！

とうけい！とはその名の通り統計学の勉強会です。大学生が教養の時に学ぶ普通の統計学をもう一回勉強するために開かれました。「やっぱこれからの時代統計は必要だよなー」ということで夏休みが終わる9月30日の小さな飲み会で日本酒片手に決まりました。

勉強会のメンバーは大体5人ほどです。会場は部室です。

使っている教科書は、東京大学教養学部統計学教室の『統計学入門』*6です。大学の講義などでも使われている標準的な教科書ということで今回使うことになりました。電子化のご時世ですが、各参加者が一冊ずつ購入して勉強会に持参しています。

スタイルは、最初の一時間で本を読んで次の一時間で討論をするという形です。討論というのは、教科書を読んで納得いかない箇所や疑問点を挙げ、お互いに意見言ったりツッコミをいれる、というものです。たとえば、この概念は何を言わんとしているのか、この数式の意味は何か、などについて「分からなかった」と言います。すると、「ここはこういうことだよ」とか、「いやそれはちがくてこっちが正しいと思う」とかの話し合いになるのでそうした話し合いを各々が納得いくまで続けます。こうすることで理解が深まります。少人数だからこそできるスタイルだと思われ

*6東京大学教養学部統計学教室、東京大学出版会、1991、ISBN 978-4-13-042065-5

ます。

よくある勉強会のスタイルは輪読で、発表担当者が予習してスライドをつくり発表する、というものですが、とうけい！はこの形式を採用していません。スライドをつくるのがめんどくさいからです。

12月1日現在、5章まで読み終わっており、今年度までに終わらせる予定です。

ちなみに、とうけい！の名前は某放課後なアニメーションになってつけられました。放課後に高級お菓子などを出すかもしれないと宣伝をしてみれば「お菓子と聞いて」とお菓子目当ての参加者が10人ほど。すでに勉強会は4回ほどやりましたが「お菓子と聞いて」の人で実際に参加しているのは1、2人。まあそのようなものなのでしょう。ちなみに今度放課後にこっそりお菓子を食べてみようかと思っています。

koji

Rails3 で Web サービス

「Web サービス設計に興味のあるすべての学友は Rails3 で Web サービスプロジェクトへ!!!!」

上記の掛け声で集まった新入生と上回生たちは、最近流行りの Web アプリケーションフレームワーク Ruby on Rails を使って効率的に Web サービスを開発する術を身に付けていくのでした。

オブジェクト指向言語 Ruby、Web フレームワーク Rails、そして問題の発見と解決過程を学ぶ勉強会です。クラウド化が叫ばれユビキタス社会が到来する中で、これからのアプリケーションの主流は Web サービスへと集約されていくでしょう。様々な Web フレームワークが乱立する中で、Rails はその正当性と生産性の高さで確固たる地位を築いています。「Don't Repeat Yourself——同じ事を繰り返すな」や「Convention over Configuration——設定よりも規約」「資源中心の RESTful な Web アプリケーション設計」を耳にした覚えはありませんか。まったく、これらの精神こそが Rails の高い生産性を支えているのです。勉強会では座学でこれらを学び、その後実際にコーディングすることで経験を積みました。また、リバタリアン・パターンリズムなど技術と人間の付き合い方についても少しだけ Tips として紹介した気がします。

教科書は、『たのしい Ruby』*7、*Ruby on Rails Tutorial**8、『Ruby on Rails 3 アプリケーションプログラミング』*9などを使用させて頂きました。

@hidesys

*7 高橋 征義・後藤 裕蔵・まつもと ゆきひろ、ソフトバンククリエイティブ、2010、ISBN 978-4-7973-5740-0

*8 Michael Hartl. <http://ruby.railstutorial.org/>

*9 山田 祥寛、技術評論社、2011、ISBN 978-4-7741-4663-8

DTM 練習会 2011

DTM 練習会 2011 担当の hideya です。この練習会は我々 KMC が新入生の歓迎・定着のために行う「新入生プロジェクト」の中の一つであり、その名の通り PC を用いた作曲 = DTM を初心者・経験者問わずみんなで練習しようというプロジェクトです。

DTM をこれから始めるという人も対象に含まれるので、まずは「こんな曲が好きだ/作りたい」というものを持ち寄って皆で聴く所から始め、作曲ソフトの基本的な使い方を学び、単音のメロディを書き、伴奏やベース、ドラムパートを付け加えていったり.....と、ゆっくりと順番にやっていきました。そして7月頃から、私が思い描いていた「練習会」の姿である「与えられたお題に従って作曲をしてきて、それに対して皆で意見を出し合って磨き上げる」という段階へシフトしていきました。

現在もこの練習会は続いており、定期的に作曲をする事によって皆のスキルは向上しているように思われます。今年は特に DTM を志そうという者が多く、しかも皆とても優秀で毎回彼らの成長を見るのが楽しみです。ていうか既に私が一番下手な気がウボァー

というわけで、これからも我々は作曲活動を続けていきます。もし皆様に我々 KMC が制作したゲームを遊んで頂ける機会があれば、我々が作った(であろう)音楽にも耳を傾けて頂けると幸いです。耳からも楽しさを感じて頂く為にも、我々は日々精進致します。それでは、運が良ければまたお会いしましょう。

いささか修行不足の hideya

とびうおのリア充物語——沖縄編 1

@tobiuo12730

沖縄へ

「花と水とパインの村 東村」——東村役所のサイトから

「あんなに女の子について情熱的に話しているとびうお氏を初めて見た」——同級生、飲み会で

僕が兼部している農業系のサークルは、毎年春と夏に沖縄にある倉庫に泊まりこみ、現地の農家さんを手伝うというツアーを企画しています。僕が比嘉さん（仮名）と知り合ったのはこの春先のことで、彼女は現地の農家さんの娘さんです。父親さんが僕らに「娘のバンドのライブを見に来てやってくれ」とチケットを渡してくださったのがきっかけで、おとなしい性格に似合わず身長よりも大きなエレキギターを弾いている様子を見て、カッコイイなあと思っていました。

2月頭には「あの子は京都に来れば絶対に売れる！」なんてことをしゃべっていましたが、そのあとしばらく比嘉さんのことは忘れていました。それが、6月に友達と飲んでいたときにいろいろあって比嘉さんの話題がのぼり、比嘉さんにメールを送ることになりました。僕が比嘉さんのアドレスを知らないと言うと、友達がこんなメールを送りました。

お久しぶりです！ とびうおっていう人覚えてる？ 彼が比嘉さんのライブ聞きたいって言うてるから、ライブの日程教えてあげてよ！（とびうおのアドレス）@softbank.ne.jp

女の子にメールを送ったことのない僕でもこんなメールアドレスの聞き方は姑息だとわかっていて、返信が来るはずはないと確信していたのですが、翌日になってみるときちんとメールが返って来ました。専門の実験などで滅入っていた時期でもあったので、僕にもようやく春が来たのではないかと大喜びしたものでした——。この頃に僕は沖縄に行く決心を固めました。



沖縄へ行くまでの間にも彼女にアタックをかけるべきだと周りから促され、友達を農学部総合館の中庭に召喚しメールの推敲をしたり、彼女が好きな RADWIMPS を TSUTAYA で借りてきたり、デスクトップの背景を僕の背中に「とびうおさんへ」と書かれたサインにしてみたりしました。上の空のうちに時は過ぎ、沖縄に出発する 8 月の頭が来たのでした。

再会

「比嘉さんを内地に連れて帰れよ」——先輩、飲み屋で

「ロック好きになるよう自分を変えてまで、その子と付き合いたいのでしょうか」——後輩、部室で

「お前が彼女を作りたいのはわかった、けどなんで沖縄やねん」——先輩、京阪電車で

やり残してしまった産業微生物学のレポートを書きながら那覇空港へと飛び、400 円の安宿に泊まり、路線バスで太平洋を眺めながら名護へ向かい、農家さんに軽自動車で拾ってもらって東村へ。ちょうど僕らのサークルと同じような目的で来ていた早稲田大学の人たちと、宿舎である倉庫で合流しました。

彼らの目的は農業体験でもありますが、さらなる目的はウークイ（旧盆 15 日）で催されるエイサーに参加することでした。沖縄では各集落の青年会の若者たちが先祖の霊を祀るためにエイサーという盆踊りを踊りますが、僕がお世話になる集落には適齢の若者が 6 人しかいないために、外から踊り子を呼ぶ必要があるようです。朝 8 時に畑に向かい、19 時からエイサーを踊り、22 時から酒を飲み、26 時に飲み直す——こんな毎日が続きました。

そんな中、比嘉さんと会う機会は訪れます。彼女にはメールで既に「おみやげを持って行きます」と言ってあったので、僕はケーニヒスクローネのミニパイセット（2,600 円）を持っていました。僕は比嘉さんの父親さんに「会って直接パイを渡したいです」と言いました。父親さんは「この時期娘はライブの練習で本部（片道 40 km ぐらい）にいるから、仕事上がりに家に行ってもまだ帰っていないかもしれない」と答えられました。実際に仕事上がりに家に行くと、誰もいませんでした。このまま父親さんに渡すのかなと思っていたところ、父親さんが「帰りがけに宿舎の倉庫に寄って行くよう言ってあげよう」と言って下さいました。

宿舎に戻り、再会の時用に三宮 VIVRE で買ってきたポロシャツ（2,000 円）やユニクロのジーンズ（4,000 円）、ZARA のローテクシューズ（定価 12,000 円）に着替え、わくわくしながら倉庫で待っていると、一台のバンが敷地内に入って来ました。既に前回会った時から半年が経過していて僕は比嘉さんの顔を覚えていなかったもので、いったいどんな子だったかとドキドキしていました。バンから制服姿の女の子が降りてきて風貌をみると、そりゃ僕は惚れてまうわみたいな気分がしました。

「ひ、ひ、ひ、比嘉さん？ こ、こ、これ、おみやげです!!」だいたいこんなかんじでおみやげを渡しながら、夏の始まりを感じました。

特集: KMC 部員と Twitter クライアント

KMC 部員の Twitter クライアント (@c_possum)	14
とびうおのリア充物語——沖縄編 2 (@tobiuo12730)	16

KMC 部員の Twitter クライアント

@c_possum

今や世間でかなりの知名度を誇る Twitter ですが、KMC 内でも Twitter の利用率は以前から高く、現在の主要現役部員 40 名程度の内、Twitter アカウントを持っていない部員は片手で数えられる程度です。

そこで部員に Twitter の利用環境を聞いてみました。今回は「home timeline(いわゆる TL)を普段どのようなクライアントから見ているか」を聞きました。17 名の部員の協力を得て、次のような結果となりました。

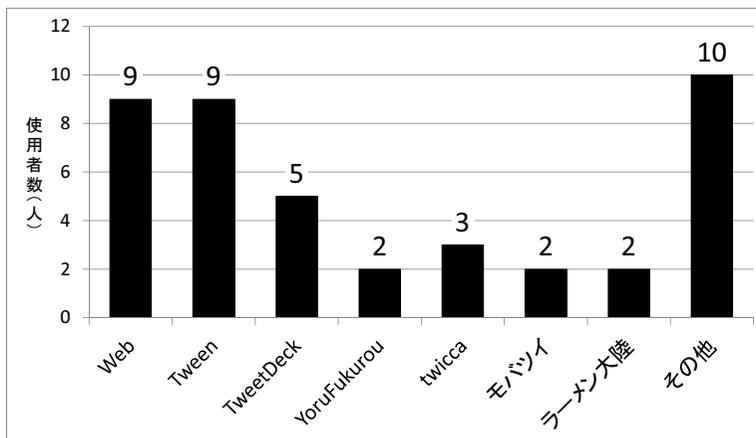


図 1 使用クライアント勢力図

なお、その他には

- HootSuite
- Janetter
- PeraPeraPrv
- Saezuri
- Tweet Caster

- Twit
- Twitter for iPad
- Twitter for iPhone
- ツイタマ
- ついっぶる

が挙がっていました。

Web、Tweent が二強、TweetDeck が次いでいますね。Twitter ヘビーユーザの多いKMC といえども、多くの方は Web を利用するようです。Tweent、TweetDeck は一般にもユーザの多いクライアントですね。納得の上位だと思えます。それ以下はばらけていますが、携帯端末でのクライアント選びがその要因の一つだと思えます。

また、自作のクライアントを用いているか聞いたところ、4 名から肯定の回答がありました。大まかな特徴を掲載します。

- Tweent 0.9.7 を複数アカウントに対応させるためのプロキシ。Ruby。
- ローカルの画像を Tumblr にアップロードし、その時に URL を tweet する。
- Haskell で書いたものと Python で書いて GAE で動かしている。
- Python と curses で作った TUI の Twitter クライアント (tuitwi^{*1})

ここに挙げられているような、Twitter クライアントのようなツールに分類されるアプリケーションは、ゲームと比べて公開されているものが非常に少ないというのが KMC の現状です。これについて、以前 OB から「部員はもっと github に登録するべきだ」という意見を頂いたことがあります。

ツールは個人か部の中で制作から運用までが閉じてしまうことが多いので、公開というステップを踏むのをためらうことが多いのでしょうか。その観点から、今年度開催されていた「Ruby on Rails 勉強会」は、ツールではなく Web サービスと呼ぶべきではありませんが、ゲーム以外の公開されるアプリケーションの勉強会として一線を画しているように思えます。部員の作った種々雑多、玉石混淆なツールが世界に公開される未来も楽しそうです。

主要な連絡は IRC が用いられる KMC ですが、雑談その他コミュニケーションについては Twitter が IRC に匹敵する程度の手段になっているように思えます。サーバの設定などが存在すること、Twitter や Skype・Windows Live メッセンジャーなどと比べて使っている人が少ないことなどの理由から、新入生への IRC の導入は障壁が高いという意見も聞こえる現在ですが、部内にユーザが未だ多い・部内に IRC サーバが存在することなどから、しばらくは IRC が主要連絡手段であり続けると思えます。基本的に公開姿勢である Twitter が基幹となることはないでしょう。

SNS の動向に合わせて、連絡でないコミュニケーション手段は移ろいで行くと思われるので、今後何が主たるメディアとなるか見守って行きたいと思えます。

*1<https://github.com/seikichi/tuitwi>

とびうおのリア充物語——沖縄編 2

@tobiuo12730

夏祭り

「大輔くん比嘉さんのこと好きなんだねー。会ってたとき汗ダクダクだったからバレバレだよ」——再会后、ツアーの女の子

「あれ、さっきポロシャツ着てましたよね？ なんで着替えなおしたんですか」——再会用の服を脱いだ後、ツアーの男の子

「通るがな通てい 自由ならんありば 神仏ていしん 当ていやならん さー思やが来ん かなしが来ん^{*1}」——エイサー「いちゅび小節」

勉強して、バイトして、恋をして——早稲田大学の人たちはそんな生活を送っているようです。昔放映されていた「あいのり」という番組を思い出させるような彼らと飲んだり踊ったりしながら、8月は順調に過ぎて行きました。8月頭に台風が来た影響で比嘉さんの家の小屋が壊れてしまって我々子どもが修復に駆り出されたこともあり、比嘉さんの家にお世話になることもたびたびありました。それでも、娘さんはライブの練習で忙しいらしく、彼女に会えたのは一度ぐらいのものでした。

家にお邪魔したとき、比嘉さんの父親さんに「早稲田の人たちが帰ってしまったあと、あの倉庫に一人で過ごすのは寂しいので、ホームステイさせてください」と切り出してみました。おみやげを渡した時点で父親さんには僕の気持ちがバレバレだと思っていたのできっと断られるに違いないと思っていたのですが、あっさりOKされてしまいました。その後数日したあとに「もしかして比嘉さんの父親さん僕の気持ちに気づいてはらへんのかな」と早稲田の人に聞いてみると、「比嘉さんが『大輔は娘目当てでうちに泊まりに来るんだ』と仰っていたよ」と言われました。やっぱりバレバレでした。ちなみにこの頃には既に僕の噂は集落全体に広がっていたようです。

旧暦7月15日はウークイと呼ばれ、道ジュネーという行事が行われます。これは各集落の神聖な場所(墓など)および祝い事があった家(新築、出産など)を巡礼し、各地でエイサーを踊るという行事です。午後6時から始まり午前1時に終わる激しい行事ですが、地元の小中学生は苦もな

*1通う限り通って、自由にならないのであれば、神仏というのも当てにはならない。

く踊り切ります。この行事をなんとか乗り越えたあと、早稲田の人たちは徐々に現地解散していきましました。

ワークイが終わったあとは、東村の夏祭りが待っていました。ちょうどその祭りでもエイサーを踊るといことになり、僕と早稲田の人1人が参加することになりました。

夏祭りでは比嘉さんがライブ演奏をするということを知っていたので、僕は夏祭りの前日に名護市までわざわざ向かい花束(600円)を購入してきていました。これをライブ後に渡す予定だったのですが、当日になってライブの時間(5:30)とエイサーの時間(6:30)が用意を含めるとダブっていることに気が付きました。仕方が無いなあとかと思いつつ、エイサーの衣装を着ながら花束を持ってうろうろしていると、地元の小学生から「なに花束なんて持ってるのー、うけるー」「その花束で締太鼓を叩くんですか」と言われました。エイサーを踊る10分前になったとき、気を引き締めていこうと、青年会のみなさんと円陣を組みました。

「みんな頑張ってください！大輔くんはその後を頑張りましょう」

僕はその場でヨロヨロと座り込みました。

エイサーを踊り終え、片付けをした後、僕の手元には花束がまだ残っていました。地元の中3の竜くん(仮名)にそのことを相談すると、夏祭りの最後には沖縄最大級の花火大会がある、お勧めのスポットがあるからそこで渡せばいいんじゃないかと勧められました。それはオシャレですばらしいと思い、さっそく三宮VIVREのポロシャツ、ユニクロのジーンズ、ZARAの靴に着替え、夏祭り会場に向かい、「レストランのある建物の2階に来てよ」とメールし、意気揚々とその場所で比嘉さんを待ちました。

「はい、行きます！」とメールが返ってきたあと全く返事もなく、レストランの2階で待てど暮らせど比嘉さんは来ませんでした。最初は僕と早稲田の人と竜くんの3人で待っていましたが、途中から2人が「比嘉さんを探してくるよ！」と言って夏祭りの中へ入って行きました。1人きりになって2階で待っていると、比嘉さんではなく比嘉さんの弟さんが通りがかりました。弟さんが「なにしてんねん！」と呼びかけるのに対して僕は正直に返すわけにもいかず、苦笑いしながらただ「花火を見にきただけっすよ」と答えました。とても気まずかったです。

しばらくすると他の人たちが戻って来ました。このあたりで警備員さんがやってきて「ここは立入禁止ですよ」と言ってきました。それはないぜ竜くんと言ってみると、竜くんは「丘の上とかきれいっすよ」と答えてくれました。なにからなにまでありがとう竜くんと感謝しつつ丘を上がっていき、「あそこ立入禁止だったよ、丘の上に来て！」とメールしました。丘の上に到達し、ベンチに腰掛けて待っていましたが今度はメールの返信すらきません。そうこうしているうちに花火大会は始まってしまいました。さすが沖縄最大級ということもあり、数千発の花火が夜空に散っていくさまは非常に素晴らしかったのですが、僕はとても寂しい気分でした。

そんな最中も竜くんは「花火なんて毎年見ているからどうでもいい、それよりも比嘉さんと大輔さんを会わせるほうがおもしろい」なんていうことを言ってくれ、丘を上ったり下ったりしながらずっと探しまわってくれました。その後、花火が終わってしまい、僕が「なんでこんなにも僕はタイミングが悪いのだろう」とベンチでしょんぼりしていました。すると、携帯に知らない番号から

電話が入って来ました。もしやと思いながら、僕は電話に出ました。

「ももももももしもし？」
『もしもしー』
「ひひひひひひ比嘉さん？」
『そうですよー』
「いいいいいい今どこ？」
『レストランのある建物の前です』
「そそそそうなん？ じゃあ今からいくわー！」

レストランのある建物にまで降りていき、比嘉さんに出会い、花束をわたし、ツーショット写真を撮り、「電話番号もらっちゃいなよー！」(竜くん)と言われ電話番号を交換し——夏祭りの次の日から比嘉さんの家にホームステイすることが決まっていたので、最後に

「明日からホームステイなのでよろしくね」

と言いました。実り多き夏祭りでした。

ホームステイ

「好きな女の子の家にホームステイしに行くって、本当にそんなことしていいと思ってるの？」——ツアーの男の子、エイサーの練習中に
「えー、好きな人の家にホームステイするなんて絶対たのしいでしょ！」——ツアーの女の子、宿舎にて
「元気だしなよ」——地元の中3「竜くん」、レストランの2階で
「比嘉さんのお父さんもよくそんな男を泊めますよね」——青年会の新婚「アグー*2」さん、宿舎で

かつて携帯電話が普及していなかった時代は、女友達と電話するには親のスキを見計らうという困難が待ち受けており、それを経なければ話すことすらままならなかったようです。それに比べると僕の置かれていた状況は理解できないくらいに好条件で、その状況を一番理解できていなかったのは僕でした。

比嘉さんの一家は母親さんがとても強い感じの人で、ナヨっとしている僕を常に弄っていました。昼ごはんのときに母親さんが残り物のシチューを食べていると「これ娘が作ったシチューやねん、たべたいやろ!!」などと、父親や弟がいる前で堂々と僕に言ってきました。父親さんは「昼間から酒盛りやってるみたいな話題だ」などと言って微笑まれました。弟さんは苦笑いです。

弟さんは母親さんに似て、僕をずっとからかっていました。僕が彼の自由研究を手伝ってあげている最中、僕が「今晚は(娘さん)と話す機会作ってくれよ」と言ってみると、弟さんは母親に「ね

*2沖縄原産の豚の一種で、みんながこんな呼び方をするので本名を忘れてしまいました。

え、とびうおさんがねえねえ貸してやって!!」などと言ってました。母親さんは苦笑いです。

弟さんはなんだかんだで家で遊んでいることが多かったですが、姉さんのほうはライブの練習で夜にしか帰ってきませんでした。しかしその頃には僕は父親さんに勧められる泡盛で酔っていて、大概の場合寝ていました。

4泊5日のホームステイのうち3泊をこのような形で無駄にしてしまい、このままではいけないと思いました。最後の晩は比嘉さんが廃鶏*3の丸焼きを用意していただき、僕はそれを食べながら父親さんと歓談していましたが、お酒のほうは「今日は若い人と喋りたいんです」と言って断りました。さて、その日は娘さんの帰りが早かったため、待ちに待った比嘉さん姉弟との団欒でした。正直いままで女性と話したことさえ少ない僕には女の子に話しかけるための話題なんて思いつかず、仕方がないので茶の間に流れているテレビを話題にしようと思いました。その時に流れていた番組が

『ザ・ベストハウス「世界のスターカーワースト3」』

でした。僕は頭が真っ白になってうずくまりました。なんというタイミングだろう——このようにして、僕のホームステイ生活は終わりを告げました。



*3卵を産み終わったメスの鶏で、この集落では祝い事や賭け事のネタとしてしばしば登場します。

部員のコラム

Haskell Golf 攻略 (@henkma)	22
はじめに	22
Anarchy Golf の紹介	22
Golf と Haskell	23
Haskell Golf の基本	24
基本戦略	25
リスト	27
思考方法・方針紹介	29
おわりに	34
【禁書 SS】インデック Scheme (@l_possum)	35
不動点の話 (@t_uda)	40
ひでシスの多様性物語——インドネシア編 (@hidesys)	45
とびうおのリア充物語——沖縄編 3 (@tobiuo12730)	50

Haskell Golf 攻略

@henkma

はじめに

本コラムでは、Haskell Golf の魅力およびそのコツについて紹介します。時間とスペースの関係で、ある程度 Haskell に関する知識を前提とします。

想定する知識・技能はだいたい以下のようなものです。

- Haskell で簡単なコードを書くことができる。
- Prelude の関数などはある程度理解している。
- `>>=` 等を使った演算と `do` 記法の書き換えはなんとなくできる。

ここでいう Golf とは、ショートコーディング、すなわち与えられた仕様を満たすプログラムをなるべく短く書く競技です。筆者は Anarchy Golf^{*1} (通称「あなごる」) というサイトで時々遊んでいます。競技プログラミングについては『独習 KMC vol.1』にて kirita 氏がいくつか、Golf については、eha 氏が C について紹介していますが、Golf は特に何の役にも立たないという点で特徴的です。

では何が良いのかと言うと、何が優秀かという定義がはっきりしていること、つまり仕様を満たして一定時間以内に処理が終わるのであれば、一打でも少ないことが他の全てに優先するということです。また、競技としてはパズルに近く、頭をひねって考えることが多いというのも良いところです。

本コラムではあなごるを前提に話を進めますが、他の Golf でも役にたつこともありますし、もしかしたらあなたの通常の業務に役に立つことがあるかもしれません。

Anarchy Golf の紹介

Anarchy Golf (あなごる) では、たくさん問題が出題されています。プレイヤーは好きな問題を選んで好きに挑戦することができます。登録やチャット、掲示板などというものはなく、ただ名前

*1<http://golf.shinh.org/>

を入れてコードを submit する。それだけです。

問題の内容は単なるプログラムの問題としては簡単*2なものが多く、いかに打数（プログラムのサイズ）を削るかに特化しているようなものが多いように思います。

問題の形式ですが、入力が 0~3 個、出力が 1~3 個あるだけという単純なもので、とにかく入力と出力がマッチ*3していれば何をやっても OK という作りです。

アルゴリズム的には怪しくてもとにかく出力があれば良いので、難しいことは考えなくても挑戦できます。

Golf と Haskell

Golf はいろいろな言語でできます。なぜ Haskell で Golf をやるのか？ それは Haskell は Golf してもそれほどコードが崩れず、本質的*4な部分で遊べるからです。

例を出します。「標準入力の奇数番目の行を表示せよ」というだけの問題です。

perl

```
#!perl -p
<>
```

なんと言えば良いのでしょうか。これはプログラムなのでしょうか？

sed

```
n;d
```

……。

例が悪いのでしょうか？ 良く使われる言語であるところの C ではどうでしょうか？

C

```
main(i){for(;gets(&i);gets())puts();}
```

だいぶプログラムらしい*5ですね。しかしよく見ると、gets も puts も引数がなんだか変です。これはスタックに何か*6が乗っていることを前提としているコードなのです。

*2FizzBuzz レベル。

*3入力 0 のものは出力が 1 つだけあってそれを出すだけです。

*4人によって全然意味違いそう。

*5型が書かれてないのは Golf だから仕方ないです。

*6この場合 &i。

Haskell

```
m@main=g>>=putStrLn>>g>>m
g=getLine
```

Haskell の例です。特に凝ったところもなく、大変素直で美しい^{*7}コードですね。

このように Haskell では Golf してもコードの構造は保たれ、プレイヤはプログラムの構造に専念してプレイすることができます。

Haskell Golf の基本

本節では Haskell Golf (あなごる) で気をつけるべき基本的なことを説明します。

基本中の基本

- 余計な空白や括弧は削りましょう。
- 改行は LF にしましょう。
- `module Main () where` は書かないようにしましょう。
- 型はなるべく書かないようにしましょう。
- Prelude の関数はなるべく覚えておきましょう。

罨

よくありそうな罨、というか、私がひっかかった or ひっかかりそうになった罨をいくつか紹介します。

- `main` の型は `IO ()` ではなく、`IO a` です。

以下の各行は同じです。

```
main=mapM_ print[1..10]
main=mapM print[1..10]
```

- 数値リテラルの後に空白を入れる必要はありません。

以下の各行は同じです。

```
mod x 3==0
mod x 3<1
gcd 3x>2
```

^{*7}型が書かれてないのは Golf だから仕方ないんです！

- (あなごるの場合)最後の空白・改行は無視されます。
つまり、最後に適当に空白や改行を出してしまっても問題ありません。
- 引数を演算子で受けることができます。

```
f x=g div x
g d x=x'd'3+x'd'10^x'd'20
```

このコードは以下のように書き換え可能です。

```
f x=g div x
g(%)x=x%3+x%10^x%20
```

- エラーで止めてかまいません。
getLine の失敗で止まる例

```
main=getLine>>putStrLn>>main
```

パターンマッチの失敗で止まる例

```
main=f[1..10]
f(a:b)=print a>>f b
```

- パターンガードが使えます。

```
f x|a:b<-x,y<-f b,y==a|length x==9|0<1=2
```

基本戦略

本節では入出力やモナド・関数合成など基本的なことについて簡単に説明します。

入出力

入出力は基本中の基本ですが、問題によって最適なものはかなり変わります。例を挙げておきます。

入力全体を使う場合は以下の形が多いです。

```
main=interact f
```

1行毎に処理する場合、処理内容によって使い分けます。

```
m@main=getLine>>f>>m
m@main=getLine>>putStrLn.f>>m
main=interact$(>>f).lines
```

入力がない場合などは以下の形も使えることがあります。

```
main=mapM putStrLn 何か
```

前節で似た形を紹介していますが、以下の形も使える場合があります。

```
main=f 何かのリスト
f(a:b)=g a>>f(aとbを使った何か)
```

他、readLn、unlines、words、unwords 等の関数は Prelude にあって入出力に便利な関数です。

条件分岐

条件分岐では、if や case を使うより、関数引数のパターンマッチやガードを使う方が効率が良くなります。

基本的な書き換え

```
if p then a else b
f;f x|p=a|0<1=b
last$a:[b|p]
```

```
if x==0 then a else b
[b,a]!!(0^x) -- x>=0 のとき
b+(a-b)*0^x -- a,b が Num のとき
```

```
if p then s else []
[c|p,c<-s]
```

```
if p then a:[b] else [a]
a:[b|p]
```

max や min は非常に強力*⁸なことがあります。

```
\x->if x>a then a else x
min a
```

import

import は高いのであまりしませんが、たまに強力な効果を生むモジュールがあります。よく使うのは List、Numeric で、特に List は有用な関数が多いです。sort や nub は効果が強力です。List の関数も Prelude 同様、覚えておくと思の幅が広がる*⁹と思います。

*⁸特に String と相性が良いと思います。

*⁹下手に考えだすと思考の深さを犠牲にするので注意！

モナド

Haskell Golf では IO はもちろんのこと、リストもよく使います。>>= や >> の使い方、do 記法などは使いこなしましょう。do 記法と >>= では、基本的に後者の方が効率が良いことが多いですが一度変数へ入れておく必要がある時などは do も有用です。

再帰

Haskell は関数型言語ですから、当然再帰的な記述をしたくなることが多いです。Golf でもこれは同様で、よくあるようなリストを辿るような処理では再帰を使いたくなります。ただ、Haskell には map、fold、scan などの有用な関数があり、これらを用いて短縮できることが多いです。

関数合成

Haskell は関数型言語ですから、当然関数合成は便利です。Golf でもこれは同様で、(.) を使って関数を繋ぐと効率化できることが多いです。λ や関数定義でちょっと複雑な式ができてしまった時は point-free style を試してみるのも良いかもしれません^{*10}。point-free については <http://www.haskell.org/haskellwiki/Pointfree> や <http://d.hatena.ne.jp/eagletmt/20090528/1243524642> などが参考になると思います。

その他の置き換えなど

最後に、いくつかの置き換えパターン等を紹介します。

- True、False は 0<1、0>1 で書き換えられます。
- `x::Int` のような型指定は `x+0` や `abs x*11` が使えます。関数として型指定する場合は `(+0)` や `abs` が使えます。
- パターンマッチでは記述順で効率化できる場合があります。

```
f []=[];f(a:b)=何か
f(a:b)=何か;f x=x -- 型が変わらない場合
```

リスト

Haskell を含む多くの関数型言語でリストの扱いは重要なファクターになります。Haskell の場合は文字列がリストですから、Haskell Golf では特にリストに関する操作を避けて通ることができません。

本節ではリストに関するテクニックをいくつか紹介します。

^{*10}flip を使うレベルになってくると良くないようすが、

^{*11}x が非負の場合。

concat、concatMap

concat や concatMap はあまり使いません。x>>=id や x>>=f、(>>=f) の形の方が有利です。

規則性のあるリスト生成

無限リストは cycle を使ったり、l=何か:1 のような定義を作ったりすると効率良く生成できます。Haskell では無限リストから適当に必要な分だけ取りだす^{*12}というようなコードが頻出しますので、無限リストの作り方も重要です。

また、いくつかの複製を作るような目的では (>>) が有用です。

```
replicate n x -- x の n個の繰り返し、効率が悪い
[1..n]>>[x] -- x の n個の繰り返し、効率が良い
[1..n]>>[x,y] -- [x, y, x, y ... x,y]
```

規則性のないリスト生成

規則性がないリストを作って適宜値を取り出すようなことがしたい場合もあります。その場合は以下のような形の書き換えが可能です。

```
["hoge","fuga","buhya"]
words"hoge fuga buhya"
```

```
[38,56,91,48,51,60]!!x
fromEnum$"∆8[03<!!x
```

値の取り出し

"aho"!!mod x 3 のようなことをしたくなるのがよくあるのですが、これは cycle"aho"!!x の方が安いです。ただし x の値が大きくなる可能性がある時は timeout の心配があります。

sort

sort は List.sort を使うと効率が良いです。何かの条件で sort する時も sortBy ではなく sort と snd を使った方が多いことが多いです。

```
sortBy(\a b->f a'compare'f b);f x=何か
sortBy((.f).compare.f);f x=何か
map snd.sort.map(\x->(何か,x))
```

^{*12}take や (!!) を使います。

また、`import List` は重いので、場合によっては `List` 内包表記を使った方が良い場合があります。

```
import List;map snd.sort$map(\x->(odd x,x))1
[x|x<-[0,2..98]++[1,3..99],y<-1,x==y]
```

`length`

`length` は場合によっては `sum` を使った変換をした方が良い場合があります。

```
length$filter p xs
sum[1|x<-xs,p x]
```

文字列の `parse`

文字列 (や単なるリスト) の `parse` では、`takeWhile` や `dropWhile` よりも `span` を使った方が有利です。

```
takeWhile p
fst.span p
```

また、`lex` は文字列限定ですが非常に強力なので頭の隅に入れておくと良いでしょう。

思考方法・方針紹介

本節では具体的な例を見ながらいくつかの考え方について説明します。

計算

具体的な例に入る前に打数 (バイト数) の計算について書いておきます。Golf では一つコードを書いてみて書き換える、複数のアルゴリズムを考えてそれぞれ実装してみる、というようなことがよくあります。これは実際にやってみて何バイトか計測しても良いのですが、時間がかかってしまいます。

頻出する関数やイディオムの打数は覚えてしまって、それらの組み合わせで脳内コーディングまたは脳内書き換えをやってしまえばかなり時間の節約になるのでお勧めです。例としては以下の様なものがあります。

- `main` = 5 打
- `interact` = 8 打
- `main=interact$` = 14 打
- `m@main` = 7 打
- `mapM` = 4 打

- print 5 打
- putStr 6 打
- lines 5 打
- unlines 7 打
- f x= 改行も含めて 5 打

まだまだ沢山ありますが、この手の数を覚えておくと少し思考の助けになるはずです。

勝てば官軍

以下に引用するのは、あなごるの問題、ABBA です。

Problem

```
AAA to B
BBB to A
ABA to AA
BAB to BB
```

Input

```
A
AA
AAA
B
BB
BBB
ABAB
AAAABBBB
ABBAABAABABABBAAB
```

Output

```
A
AA
B
B
BB
A
AAB
BAAB
A
```

この問題、何やらアルゴリズムは複雑そうなのですが、input/output が 1 組しかありません。このような問題の場合はこうやってしまいましょう。

解答例

```
main=putStrLn"A\nAA\nB\nB\nBB\nA\nAAB\nBAAB\nA"
```

とにかく input/output がマッチしていれば良いのですから、こういうのもアリです。こういう問題は賛否両論ありそうですが、このようなものも許されるのが他の競技プログラミングとの違いですね。

理想形を考える

以下に引用するのは、あなごるの問題、ABBA2 です。問題は ABBA と同じですが、input/output が二組あります。

Input 1

```
A
AA
AAA
B
BB
BBB
```

Output 1

```
A
AA
B
B
BB
A
```

Input 2

```
ABAB
AAAABBBB
ABBAABAABABABBAAB
```

Output 2

```
AAB
BAAB
```

A

今度はまともな実装をする必要がありそうですが、実は所詮二組なので単なる条件分岐で十分です。

解答例 1

```
main=interact f
f x|x<"AA"="A\nAA\nB\nB\nBB\nA"|0<1="AAB\nBAAB\nA"
```

素直に実装しようとするこのような形になります。しかしここで考えるべきは「条件分岐では max/min が有利」という事実です。「もしかして max/min が使えないか？」と考えるわけです。ここに考えが至れば以下の解答がでてきます。

解答例 2

```
main=interact$max"A\nAA\nB\nB\nBB\nA".min"AAB\nBAAB\nA"
```

囲碁や将棋でもそうですが単に全探索的に考えるより「この形になるといいな」という感性が有利に働くことがあります。この問題はその例の一つだと思います。

リストモナドでの mapM

リストモナドですから、モナドとしてのリストに対して mapM を使うことができます。mapM は map して sequence するような関数なので^{*13}、

```
mapM(\_->"01") [1..3]
["000", "001", "010", "011", "100", "101", "110", "111"]
mapM(\_->"012") [1,2]
["00", "01", "02", "10", "11", "12", "20", "21", "22"]
```

というように、簡単に 0 filled な N 進数表現が簡単に生成できます。

応用例の一つ書きます。これは、Factradic Counter というあなごるの問題で階乗進数（一桁目は 0 だけ、二桁目は 0 と 1、三桁目は 0 と 1 と 2、……というように桁によって使える記号が増える数値表記法）で、0 から 1000 まで表記せよという問題の解答です。

```
main=mapM(print.abs.read)$take 1001$mapM(\x->['0'..x])"6543210"
```

このように、mapM は非常に強力なことがあります^{*14}。

^{*13}詳しくは Hoogle 等で実装を見てください。

^{*14}abs が入っているのは read の型を確定させる工夫です。

素数

あなごるの問題、Palindromic prime を紹介します。11, 101, 929 のような回文状の素数を 20000 まで表示せよという問題です。

あなごるではこのような素数に関する問題や約数に関する問題が多くあります。普通に素数判定しても良いのですし、素数を列挙してから判定するようなことをしても良いのですが、フェルマーテスト^{*15}という良い方法があります。フェルマーの小定理:「 p を素数とし、 a を p の倍数でない整数としたとき、 $a^{p-1} \equiv 1 \pmod{p}$ 」を応用したもので、「 n と互いに素な整数 a が $a^{p-1} \equiv 1 \pmod{n}$ ならば n は素数であろう」というものです。

これは完全な素数判定ではないのですが、比較的精度が良いので他の条件とからめることによって有用な判定となり得ます。この問題でもフェルマーテストが使えます。

解答例

```
-- 基本的なところ
main=mapM print$2:[x|x<-[3..20000],mod(2^x)x==2,show x==reverse(show x)]

-- 少し効率化
main=mapM print[x|x<-[2..12^4],mod(2^x-2)x+x==(read.reverse.show)x]

-- gcd を使うとさらに効率的
main=mapM print[x|x<-[2..12^4],x==read(reverse$show$gcd(2^x-2)x)]
```

規則性を見抜く

あなごるには数列を出力する問題が沢山あります。そのような問題では、数列の性質を見抜いて式を立ててみるのが重要になる場合もあります。

以下は、あなごるの問題、Perfect Square Free です。

Problem

```
output all the numbers from 3 to 100 without perfect square.
```

Output

```
[3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, (中略), 99]
```

要するに平方数でない数を 3 から 100 まで出せば良いのですが、ここで「平方数は規則的に並んでいる」ということに着目できれば $f(n) = n + \lfloor \sqrt{n} \rfloor$, $n \in \mathbb{N}$ というような式が浮かんできます。

^{*15}詳しくは Wikipedia とかを見てください。

これを少し調整すれば、以下の解答が得られます。

解答例

```
main=putStrLn$show[round$x+sqrt x|x<-[2..90]]>>=max", ".(:[])
```

このように良く見ると規則性のある数列になっている問題は結構あります。数列の問題を見た時は少しこういった方向にも思考を巡らせてみると良いです。

おわりに

いろいろ書いてきましたが、Golf は何ととっても根性だと思います。忍耐強く最適解を探す根気が何より必要です。

以下@付きは twitter ID です。

@shinh さん、素敵なおサイト(あなごる)運営ありがとうございます。@notogawa さん、@Lost_dog_ さん、@eagletmt さん、ブログの記事等を勝手に参考にさせていただきました。勝手にありがとうございます。@ikegami_ さん、@shelarcy さん、いつも Haskell についての情報ありがとうございます。校正とチェックをしてくださった KMC の皆さん、特に細かいチェックをしてくださった @atmonad さん、ありがとうございます。全ての Haskeller の皆さんありがとうございます。

最後に参考文献です。

- 「ぼくのぬまち出張版」@notogawa さんのブログ <http://d.hatena.ne.jp/notogawa/>
- 「いじわるだねっ」@Lost_dog_ さんのブログ http://d.hatena.ne.jp/Lost_dog/
- 「EAGLE 雑記」@eagletmt さんのブログ <http://d.hatena.ne.jp/eagletmt/>
- 「Hoogle」Haskell 用の検索サイト <http://haskell.org/hoogle/>

【禁書 SS】インデック Scheme

@l_possum

「あれ？ とうま、今日は宿題やらないの？」

「やってるだろー。」

同居人のシスターから発せられる遊んでオーラを受け流しつつ、上条当麻は目の前のノート PC のキーボードから目を離さずに答えた。

「だって大きいパカパカで遊んでるだけに見えるんだよ。」

「これが本日の上条さんの宿題なんですよー。えーっと m、m は……。」

この日、上条は学校で課せられたプログラミングの課題をこなしていた。

課題の内容は、配列も登場しないような初歩的なもので、言語は C が指定されていた。

パソコンを持たない上条は学校の貸し出し用端末を借りて、もちろんブラインドタッチなどできるはずもなく、キーの印字を探しながら人差し指で押して行くという初心者丸出しのタイピングではあったが、既に課題の終わりが見えていた。

「よーし、宿題終わりっと。」

「終わった？ じゃあとうま遊んでよ！」

「そーいやインデックス。」

「なあに、とうま。」

「お前、いい加減こっちの物の名前を覚えたらどうだ？ ケータイだってパカパカって呼んでるままだろ。」

「パカパカって言ってもちゃんと通じるし問題ないんだよ。」

「ほほう、何万冊も本を覚えてる動く図書館ともあろうインデックスさんらしくないですねー。」

「むう〜〜!! そこまで言われたら黙っているわけにはいかないんだよ。その大きいパカパ……。」

「パソコン。」

「ば、ばそこん！ だって使って見せるんだから！」

「ほーう、じゃあインデックスさんの今後の努力に期待って事にしておいてやるよ。」

「ぜーったいバカにしてるね？ いいよ、そんなこと言っていられるのも今の内なんだから！ もう今日は寝るもん！」

「はいはい、おやすみー。」

(あんまりこっち側のことを知らないのもあれだから良い機会だろ。どうせすぐ飽きるだろうし。)

翌日の昼、同居人の置いていってくれた昼ご飯を食べ終えたインデックスは、机の上の「大きなパカパカ」に顎を乗せて途方に暮れていた。今頃懸念の原因の男子は学校で弁当を食べている頃だろう。

「うー……、とうまにあんなこと言っちゃったけどどうしよう。わたしの本の中にもそんなの無いし……。そう言えばこれってとうまの学校の宿題……、こもえ！」

日頃上条の帰りをただ待ちわびている居候は、今日ばかりは昼下がりのまったりタイムを返上して、上条の通う学校を一路目指した。

「こもえー！」

職員室で昼食後に一服していた月詠小萌の穏やかな昼休みは、勢いよく開いたドアの音と自分を呼ぶ少女の声に打ち破られたのであった。

「ふわわっ!? シスターちゃん、学校に部外者が勝手に入って来ちゃダメなのですよ！」

啞えていたたばこを携帯灰皿につっこみつつ突然の侵入者の方を向くと、

「こもえ！ ばそこん教えて！」

「ふえ？ パソコン、ですか？」

「うん！ とうまが使ってた大きいパカパカのことなんだよ！」

聞けば、教え子が宿題をやっている姿を見て自分も学びたくなったと言うことらしい。

「つまり、シスターちゃんは学びたくてわたしの所に来たのですね……。教師たる者、学ぶ意志を無碍にするわけにはいかないのです。いいですよシスターちゃん！ この月詠小萌がばっちり教えてあげます！ そもそも学校の情報教育は貧弱だと思うのですよ。これだけコンピュータが身の回りに溢れているのに、その中身の初歩的なことも知らないだなんて間違ってます！」

(な、なんかよくわからないけど頼りにはなりそうかも……?)

「上条ちゃんがやっていたのはきっとプログラミング基礎の宿題ですよ。先生はその担当じゃないので内容はよく知りませんが、学ぶ意志があるシスターちゃんにはこの本を教科書にお薦めするのです。」

そう言って、インデックスに B5 程度の大きさで紫色の分厚い本を手渡した。

「*Structure and Interpretation of Computer Programs* ?」

「そうです！ 邦題は『計算機プログラムの構造と解釈』ですね。この本はコンピュータプログラミングの基本的構造をひもとく、言わばこの分野の原典とも言えるべき本なのですよ。」

「原典!? 学園都市にも魔導書みたいな本があったんだね……。」

「魔導書？ ああ、確かにそうかもしれませんね。その本の始めの方を読むとそんなことが書いてあるのです。」

「こもえ、これ読んでも大丈夫な本なの？」

「？ シスターちゃんが読み終わるまで貸してあげますから、気にしなくていいですよ。あっ、そ

れと課題を解くのに環境構築が済んでないとダメですよね……。シスターちゃん、パソコンは持ってますか？」

「わたしは持ってないけど、とうまが使ってたのはあるよ。」

「上条ちゃんのってことは……学校のですから、大丈夫ですね。ここでやり方だけ教えちゃいます。えっとですね、この本の中には練習問題がいっぱい出てくるのですよ。これを読んで勉強するには、問題を解くのが一番ですから、問題を解くときに使う道具だけ教えておきます。シスターちゃん、ちょっと隣に来てください。」

「うん。」

「このパソコンは上条ちゃんの使っているのとだいたい同じ物なので、使い方は一緒です。電源はここですよ。起動したらアプリケーションから……。」

数時間後、Scheme 処理系とエディタの使い方を覚えたインデックスを送り出す小柄な教師の顔には、教職の信念を全うしたという満面の笑みが浮かんでいた。

その後数日、インデックスと飼い猫のスフィンクスによる昼下がり SICP 読書会が開催された。

「このパカパカに精霊が入ってるの？ こもえはこの街のたいていのものにコンピュータが入ってるって言ってたし、学園都市って魔術だらけ？」

「ぶるぐらみんぐって呪文詠唱と一緒になんだね。それならわたしの得意分野かも！」

「あれ？ じゃあとうまは魔術師……？ 学園都市は普通とは違う方法で魔法を使えるような方法を作り上げてたんだね……。」

「この文字のポチポチが苦手かも……。声で言うこと聞いてくれたらいいのに……。」

「だあああ!! わかるかあああ!!」

キーボードのまばらな打鍵音が響くだけだった部屋に怒号が響いた。

「挿入ソートとかわけわかんねーよ……なあ、スフィンクス、お前やってくれよこの課題。」

ヘルプを求められたネコはイルカに聞けとでも言いたげにあくびを一つしただけで、膝から降りて歩いて行ってしまった。

「あー、もうこのまま寝れば……いやいや居残りが……。」

「とうま？」

「なんだーインデックスー。晩飯はさっき食っただろー。」

「そんなことはわかってるんだよ！ どうしてとうまはスフィンクスに助けを求めてわたしには助けを求めないのかな!？」

「助けてもらえそうもない奴に声かけたって無駄だろ。」

「それはスフィンクスより助けにならないって思われてたってことかな!? もー、ちょっと宿題見せて！」

「ほいよ、これだけ。」

「ただるそうに宿題のプリントを見せるといらだちを隠せない少女は、
「これをプログラムで動かせば良いんだね？ パソコン貸して。」
「はあ？ 借り物なんだから壊すなよー。機械音痴のお前に何が……ってええっ!?」
承諾するまもなく上条から PC を奪って自分の方に向けると、軽快にキーボードを叩きはじめた
ではないか。
「あのインデックスが……食券も買えなかった奴が……キーボードを華麗に叩いている……？ こ
れは、いったい、どういうことですか？」
数分後、なにかがぶつぶつつぶやきつつキーボードを叩いていたシスターは、自信満々の笑みで、
「はい、とうま。これでいいんだよね？」
啞然として固まったままだった上条に PC を差し出した。
「ちゃんと並び替えられてる……。すごいぞインデックス!! いつの間にこんなスキル身につけた
んだ!？」
「わたしは約束をちゃんと守るんだよ!!」
「すげー、ところでインデックスさん？ このプログラムのコードはどこに？」
「ん？ はい、これだよ。」
エディタに切り替えられた画面を見た上条からは、それまでの欣喜雀躍ムードは消え失せて
いた。
「……あの一、インデックスさん？」
「なあに？ とうま。」
「これは何語であらせられるのでしょうか？」
「これは Scheme だよ。Lisp の方言の一つだね。」
「……は？」
「知らないの？ Lisp って言うのは 1950 年代後半に、ジョン・マッカーシーが考案した言語で、
元々はプログラミング言語としてじゃなくて数学で使うモデルとして設計されてたんだよ。でも考
案されたすぐあと……。」
「ナンデコンナ二括弧多インデスカ……？」
「それはそう言う形式の呪文だから仕方ないんだよ。」
「……。」
「どうしたの？ とうま。」
「そうだよな……俺にそんな都合良く幸運なんて舞い降りるはず無いよな……。」
全ては幻想だったのだと悟った彼の気を知ってか知らずか、一仕事成し遂げたインデックスは立
ち上がって胸を張り、上条を横目に、
「ねえ、とうま。助けてあげたこのわたしに何か言うことがあるんじゃないかな？」
「え？」
「機械音痴って言われ続けてたわたしがこんなにがんばったんだから、なにかしてくれてもいいん
じゃないかな？」
自分にとってまったく役に立たなかった仕事に、言外にご褒美を要求する同居人を目の前に

して、

「.....不幸だ。.....不幸だああああああああああ!!!!」

今宵も学生寮に不幸な男子高校生の雄叫びがこだまするのであった。

おわり

不動点の話

@t_uda

この記事では、数学の様々な分野で現れる色々な種類の「不動点」を、プログラミングや理論計算機科学との関連なども踏まえつつ紹介していきます。

不動点とは

不動点 (fixed point)^{*1}とは、ある変換で動かない点のことです。厳密には、写像 $F: X \rightarrow X$ に対して $F(x) = x$ なる点 x を F の不動点と言います。具体例を見ていきましょう。

例. 線型変換と平行移動

拡大や回転といった代表的な線型変換の不動点はどうなるのでしょうか。まず、原点を中心とする拡大変換では、原点以外の点は全て元の位置よりも遠ざかるので、不動点ではありません。また、原点を中心とする回転変換でも、原点以外の点は全て元の位置から回転してズれるので、不動点ではありません。従って、これらの変換では原点が唯一の不動点になっています。一方、 x 軸に関して対称な位置に点を移す変換では、当然、鏡の役割をする x 軸が変換に関して不変です。従って、 x 軸上のすべての点が不動点になります^{*2}。

このように、不動点は一箇しかないこともあれば（無限に！）たくさんあることもあります。もちろん、一個も不動点がない場合もあり得ます。例えば、平行移動は不動点を持ちません。

不動点定理

さて、そうすると当然「不動点はいつ存在するのか？」が気になります。この問いに答える「不動点定理」には分野を超えて様々な種類のものがあります。それらを紹介しましょう。

^{*1} 固定点と訳す場合もあります。

^{*2} 一般に、固有値 1 に対応する固有ベクトル空間の元が線型変換の不動点です。

縮小写像原理

写した点と点の距離が小さくなるような写像を縮小写像と言います*3。そして縮小写像は不動点を唯一つ持つと主張するのが縮小写像原理です*4。縮小写像原理の最大の特徴は、不動点を単純な反復列で計算できる所にあります。縮小写像 F の不動点 x は、次のように適当な初期点から始めて F を繰り返し施す数列の極限で表されます。

$$\begin{cases} x_0 : \text{given} \\ x_{n+1} = F(x_n) \end{cases} \implies x = \lim_{n \rightarrow \infty} x_n, F(x) = x$$

数回反復計算を行えば、計算機でも簡単に縮小写像の不動点を近似計算することができます。

縮小写像原理の応用として、例えば Newton 法のようなアルゴリズムがあげられます。Newton 法とは、関数の零点探索アルゴリズムで、適当な初期点から始め、接線を引いてその零点を新たな近似値として反復計算するものです。具体的には、 f の零点 x は次の漸化式で定められる数列で計算できます。

$$\begin{cases} x_0 : \text{given} \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \end{cases} \implies x = \lim_{n \rightarrow \infty} x_n, f(x) = 0$$

この数列を定める漸化式の右边が x_n の関数として縮小写像になるように初期点 x_0 を上手く選べば、反復列が関数 f の零点に収束することが証明できます*5。

Kleene の最小不動点定理

次は違うタイプの不動点定理も紹介しましょう。

(D, \sqsubseteq) を半順序集合とします。つまり、集合 D 上の 2 項関係 \sqsubseteq は反射律・反対称律・推移律を満たすものとします。またこの順序に関する最小上界を \sup で表します*6。最小上界が分からない人は、単に最大値 (max) のようなものと思ってもらっても構いません。一般に最小上界は存在するとは限りませんが、任意の増大列 $A_0 \sqsubseteq \dots \sqsubseteq A_n \sqsubseteq A_{n+1} \sqsubseteq \dots$ が最小上界をもつ場合 D は完備 (chain-complete) であるといえます。

例えば、自然数*7からなる集合全体 $2^{\mathbb{N}}$ は、通常の包含関係 \subseteq で完備半順序集合になります。この時、 $A_n \in 2^{\mathbb{N}}$ の最小上界 $\sup_n A_n$ はそれらの和集合 $\bigcup_n A_n$ に他なりません。

*3 正確には、 F が縮小写像 $\iff \exists r \in [0, 1)$ s.t. $\forall x, y, |F(x) - F(y)| \leq r|x - y|$

*4 類似の定理として、Brouwer の不動点定理や、Schauder の不動点定理があります。縮小写像原理はこういった不動点定理の特別な場合と考えることができます。また縮小写像原理自体も、Euclid 空間に限らず一般の Banach 空間でもそのまま考えることができます。これらの定理は、関数空間に適用することで微分方程式の初期値問題の解の存在と一意性を示すことができるなど、数学でも非常に重要な定理となっています。

*5 もちろん、関数がそもそも零点を持たなければ収束しませんし、複数の零点があれば初期点に依存して別の零点に収束します。零点の存在や微分可能性などの良い条件があって初めて一意的な収束が示されます。

*6 この分野では最小上界の記号として \sqcup を和集合の記法と同じように用いることが多いですが、ここでは見慣れた方の \sup を使います。

*7 なお本記事では 0 は自然数とする。異論は認めない。0 は自然数 0 は自然数 $0 \in \mathbb{N}$ $0 \in \mathbb{N}$ $0 \in \mathbb{N}$ $0 \in \mathbb{N}$

完備半順序集合 D 上の関数 $F : D \rightarrow D$ は、任意の増大列に対して F と \sup が交換可能、すなわち $\sup F(x_n) = F(\sup x_n)$ であるとき連続であるといえます*8。

例えば、今分かっている素数の集合を入力として、次の素数を見つけて加えた集合を返すプログラム Prm を写像 $\text{Prm} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ だと考えましょう。この写像が連続であるというのは、無限の長さの入力（実際には計算できない！）を受け取ったときの計算結果が、有限の長さの入力を受け取ったときの計算結果で近似できるということを表しています。

さて、準備はここまでにして Kleene の（最小）不動点定理を紹介しましょう。Kleene の不動点定理は、最小元 \perp_D を持つ完備半順序集合 D 上の連続関数 $F : D \rightarrow D$ は最小*9不動点 x を持ち、更にその不動点は次のように具体的に計算できるというものです。

$$\begin{cases} x_0 = \perp_D \\ x_{n+1} = F(x_n) \end{cases} \implies x = \sup_n x_n$$

この定理を先程の素数の例で見てください。 $2^{\mathbb{N}}$ の最小元 $\perp_{2^{\mathbb{N}}}$ は空集合 \emptyset です。空の入力を受け取ったプログラム Prm は新しい素数 2 を見つけて $\{2\}$ を返します。計算を続けさせると、 $\{2, 3\}$, $\{2, 3, 5\}$, ... と次々に素数を見つけていき、最終的にその「極限」*10は素数全体の集合になるでしょう。これが Prm の最小不動点です。不動点までの増大列は「情報量が増加」しつつ「より良い近似」になることを表していて、不動点の最小性は、素数以外の余分な出力が含まれないことに相当します。

このように捉えると、Kleene の不動点定理の主張も縮小写像原理と同様に「反復で不動点を近似できる」点にあることが分かります。扱う対象や条件が全く異なるにもかかわらず主張が似通っているのは興味深いですね。

この不動点定理は計算機科学との関連も深く非常に重要です*11。次の応用例を見てみましょう。OCaml や Haskell のような関数型言語では、再帰的なデータ構造を定義できます。

```
type nat = Zero | Suc of nat;;
```

このような再帰的な構造をしたデータ全体の集合は、関数の最小不動点として定義できます*12。 Str を文字列全体の集合とします。 $F : 2^{\text{Str}} \rightarrow 2^{\text{Str}}$ を $F(S) := \{\text{Zero}\} \cup \{\text{Suc } s \mid s \in S\}$ で定めると、 F の最小不動点は $\text{Nat} = \{\text{Zero}, \text{Suc Zero}, \text{Suc Suc Zero}, \dots\}$ となります。

この考え方を一般化して、ある文法規則を満たすような文字列全体の集合は、その文法の生成関数の最小不動点として与えられることが分かります。例えば、次の BNF に従う文字列の集合 Λ も同様にして与えられます。

*8 Scott 位相での連続関数と今考えている連続関数の定義は同値になります。

*9 不動点 x が最小とは、任意の不動点 y に対して $x \sqsubseteq y$ となることです。

*10 実際、増大列の最小上界のことを極限ということがあります。

*11 類似の定理として、(Knaster-)Tarski の不動点定理もあります。条件や設定が若干異なりますが、定理の主張する所や使い方はほぼ同じです。

*12 再帰的な定義に慣れている方は「不動点なんか持ち出さなくても定義できるじゃん？」と不思議に思われるかもしれませんが、しかし、定義に自分自身を持ち出すと循環定義になってしまうため、そのような定義は数学的には決して自明なものではないことに注意してください。

$\langle \text{Lambda} \rangle ::= \langle \text{Var} \rangle \mid \text{lambda } \langle \text{Var} \rangle. \langle \text{Lambda} \rangle$

Scott の逆極限構成

文法を満たす文字列の集合（言語）が与えられても、文字列としての意味しか持たないので、その「意味」を与えたくなくなります。意味というのは、例えば先程の Nat の例でいえば、対応する自然数を与える関数 $[-] : \text{Nat} \rightarrow \mathbb{N}$; $[\text{Zero}] = 0$, $[\text{Suc } s] = 1 + [s]$ のことです。では、型なし λ 計算の意味関数^{*13}は作れるのでしょうか。つまり、（文字列としての） λ 項の集合 Λ からその λ 項が意味する関数を得るような意味関数 $[-]$ は存在するのでしょうか。

少し考えてみると、この問題は数学的にかなり難しいことが分かります。意味関数の値域として適当な「関数」の集合をとりたいのですが、この「関数」は「関数」を受け取ったり「関数」を返したりし、関数であると同時にまた値でもあることが要求されるのです。そうすると、 $D = D \rightarrow D$ （但しここで $A \rightarrow B$ は A から B への写像全体の集合）を満たすような集合を λ 項の意味関数の値域として採用すればよい気がします。ところが、 $D \rightarrow D$ の濃度は D の濃度よりも真に大きいので、このような集合 D は存在しません。そこで、写像全体の代わりに連続関数全体 $[D \rightarrow D]$ に制限すれば、領域方程式^{*14}

$$D \stackrel{\phi}{\underset{\psi}{\doteq}} [D \rightarrow D]$$

の解が構成できます（Scott の逆極限構成）。紙面の都合上、詳細は割愛します。

しかし、更に複雑な言語の意味関数を考えるなら、値域も更に複雑になります。例えば int と bool と tuple と lambda 式がある言語なら次のような方程式を解くことになるでしょう。

$$D \stackrel{\phi}{\underset{\psi}{\doteq}} \left(V_{\text{int}} + V_{\text{bool}} + \bigcup_n D^n + [D \rightarrow D] \right)_{\perp}$$

実は、Scott の逆極限構成の一般化で、圏 \mathcal{C} と関手 $F : \mathcal{C} \rightsquigarrow \mathcal{C}$ が良い性質を持てば、方程式 $D \doteq F(D)$ を満たす解 D が同型を除いて一意に存在することが Smyth-Plotkin の基本補題より知られています。これはいわば関手の不動点定理です。この基本補題より上のような領域方程式でも解が存在することが言えます。

まとめ

3 種類それぞれ異なる「不動点定理」を駆け足で紹介してきましたが、いかがだったでしょうか。もちろん、ここで紹介した以外にも様々な「不動点」や「不動点定理」があります。例えばコッホ曲線のように自己相似なフラクタル図形はある図形変換の不動点になりますし、カオス力学系で扱

^{*13} semantic function のことですが、正しい和訳を知りません。解釈関数の方が良いかもしれません。

^{*14} 最小元を持つ完備半順序集合のことを領域 (domain) と呼びます。そのため、領域 D についての方程式を領域方程式と呼ぶことがあります。なお、領域間の連続関数全体も自然に領域を為します。

うローレンツ・アトラクターなどでも不動点の概念が重要になります。また不動点コンビネータも lambda 計算における重要な考察対象です。

$F(x) = x$ という、たったそれだけの式の周りには、様々な数学と事象の本質が見え隠れしています。分野やアプローチこそ違えど、「不動点」という考え方が数学とプログラムにおいて重要な概念であることをなんとなく理解して頂けたら幸いです。

最後に、参考文献をいくつか上げておきます。

- 数値解析、一松 信、朝倉書店、1982、ISBN 978-4-320-01701-6
- Proofs and Types, Jean-Yves Girard, Cambridge University Press, 1989, ISBN 978-0-521-37181-3, <http://www.paultaylor.eu/stable/Proofs+Types>
- Types and Programming Languages, Benjamin C. Pierce, The MIT Press, 2002, ISBN 978-0-262-16209-8, <http://www.cis.upenn.edu/~bcpierce/tapl/>
- Basic Category Theory for Computer Scientists, Benjamin C. Pierce, The MIT Press, 1991, ISBN 978-0-262-66071-6
- Theories of Programming Languages, John C. Reynolds, Cambridge University Press, 1989, ISBN 978-0-521-10697-9

ひでシスの多様性物語 ——インドネシア編



@hidesys

なんの話だったっけ。そうそう、この夏にインドネシアに行った話をするんだった。

インドネシアは人口2億3千万人の国。多数の島々は東西にわたって長く分散し、厳格なムスリム・シャーマニズム・キリスト教徒・ヒンドゥー教・伝統土着信仰などの多数の信仰と種族・言語や生活方式を抱える。「多様性の中の統一 Bhinneka Tunggal Ika」を国家スローガンとして標榜する多民族国家である。政治的には、オランダ統治・日本軍政ののちの30年に及ぶスハルト開発独裁とアジア通貨危機に抛る政権崩壊・IMFによる構造調整政策など、国民の積極的自由のための基盤が整備されていなかった/いない国である。パッと見首都は発展してるように見えるけども。まあ、とりあえずだらだら書く。

ジャワ島

8/12~15は東京。コミケやな。設営手伝った。それと、ホテルで隣になった外人と話をしたら「オーウ、ワタシ slime girls 好きなんデース。」とかいうからコミケー一緒に行ってアキバ案内した。状態変化嗜好の人同士が偶然にもホテルで隣になるだなんてなかなかないのではないかな。お腹空いたからすき家行こうすき家って誘ったら、「アノ Sukiya Water はなんなのデスカ？ 飲めば飲むほど喉が渇く。」って麦茶を嫌がられたので松屋に変更したり。面白いワシントン人でした。この前お礼ってアメコミが届いたので、なにか送り返さないよ。



たしか16にジャカルタ着いた気がする。首都のジャカルタは人の数も車の数も物価も大気汚染もひどくて、なかなか住み心地が良くなかった。辛い。

18~19に深夜列車でジョグジャカルタに移動。特急に乗りたかったんだけど4500円したので350円のエコノミー列車を使った。10時間揺られたんだけど、超過満員で席は無いわ電灯はずっと



点いてるわ物売りが声を張り上げながらずっと巡回してるわ駅に止まると流しのギター弾きが入ってくるわで寝れなかった。狂ってる。労働者の置かれてる環境が悪すぎ。インドネシアでは異性装は犯罪なんだけど、スーパーマーケットで女装の人がいてほっこり。ジャカルタのホテルでめっちゃ蚊に刺されて「右手だけで 26 箇所も蚊に噛まれとるわ、ガハハ！」してたんだけど、よく見たら全身に赤いポツ

ポツが蔓延ってる事に気付いて病院へ走った。

20 に特急でソロへ移動。1 時間 100 円。列車の中で知り合った方にホテルまで車で送ってもらった。ソロはジャワの精神に基づいているらしい。ありがとうございました。まだ人とバイクの数は多かったんだけど、それなりに過ごしやすいような。リングとヨーニ（性器の象徴）が象られた神殿とか見に行く。ホントにすごい。中心部からバスで 2 時間バイクで 1 時間移動して標高 1500m のところまで行ったんだけど、神殿に着くのにな落ちたら死ぬ系の山道を歩いたりとか。



スラウェシ島

そして、スラウェシ島の真ん中ぐらにある高原都市タナ・トレジャ。

時刻表では 20 時出発で目的地に着いたのが翌朝の 7 時。下痢でおなかいいたいわ、山道でひどく酔うわ、クーラーの送風口から冷水がぼたぼた顔に垂れてくるわで寝られなかった。酔ったままホテルに着いたら、カメラと眼鏡なくしてることに気付くし.....。

タナ・トレジャっていうのは山奥の地域なんだけど、だいぶ物資は入ってきてる。市場化も進んでて街にはスーパーもあったり。でも、自分たちの伝統は大切にしていこう、っていう雰囲気と使命感を感じさせるところだった。例えば、家の形が北に向かって船みたいになってたり、葬式は何百人も人を呼んで盛大にしたりとか。



水牛と日用品の定期市は 6 日に 1 回行われる。水牛は本来色が黒いんだけど、稀に生まれるホルスタインみたいな白色の混じったのは普通の 1.5 倍の 40 万円ぐらいするらしい。肉も乳も質は同じなんだけど、ただ葬式の犠牲に使うときに良いとされてるから高いのだと。四つ葉のクローバーもそうだけど、遺伝子異常には希少価値があるのか。

あと、トレジャ人の村人の男性はみんな腰にナイフを差してた。それで稲刈りをしたり水牛を殺したり。食堂で隣りに座られたときにベンチに刀があたって「コンッ」って振動が伝わってきたり

する。ちなみに、刃は自動車の板バネを加工したもの。定期市で刃や鞘を売ってる後ろで刃を叩いて作っていたり。

スンダ列島

バリはいつもどおりのいわゆるバリ島だった。観光客でいっぱい。

最後に行ったスンバ島はインドネシアの中でも西の方で小スンダ列島の一つに数えられる、太平洋に浮かぶ小さな島。

主要な貿易圏から外れて、インドネシアの大卒初任給は1.5万円なのに、スンバ島の村人平均収入は8000円程度しかなくて、結構貧しい。西は結構湿潤で耕作に適してるんだけど、東は熱帯モンスーンの影響で乾燥してるから主要産業は牧畜。東ではまだ王政が各村に敷かれてて、奴隷制も残ってたりする。回ったのは全部で3つの街と6つの村。



いやー、11時半発やかならな飛行機。いやー11時半やかならな。ドメスティックやし1時間半前に着いといたら余裕やる。とか思いながら悠々とバイク(Ojek)でデンパサール空港に10時過ぎに着いて、セキュリティーチェックに見せようと航空券控えを出したらそこには「Departure 10:10, Arrival 11:35」の文字が……。急いで係員に問い合わせたんだけど「Late check-in.」って言われてしまった。ゆるし

ておくれ。その後少しおちこでたりしたけどそこで機転を効かせて航空券カウンターへ。明後日のフライトをBeli(購入)。1万円ほど飛んでいった。お金ないのに……。

スンバ島に着いたその日にミニバスを乗りしくじって、25人しか村人が居ない寒村に迷い込んだ。村に入ったら各家から村人がぞろぞろ出てきて取り囲まれたり。次に行った村では、囲炉裏に放り込んで焦げた表面をナイフで小削ぎとった地下茎系の芋をごちそうになった。

次の日に雇ったガイドが実は英語を全然喋れなくて、「農地の耕作権が各家族にあるならば、家系が途絶えたときに残された農地を再び割り当てる分配権は誰が持つてるの?」とかをインドネシア語で伝える羽目になった。まあ結構何とかなったので、その日以降は通訳なしで村を回ることができたけど。奴隷制の残る東地方では王様から直で話を聴けた。あなたはどれほど土地を持ってますか?って訊いたら「なんだ、キミは土地を買いたいのか。話を聞こう。」ってなってビビった。話を聴きたいのはこっちやで。



西から東への移動のミニバスで運ちゃんと話をしたら「ミスター、メモリー持ってる？」って訊かれて、USBのmp3プレーヤーを貸したら「そうよ、いたぶーる快~感 はーげーしーい炎、胸を焦がす~」みたいな歌詞が車内に大音量で鳴り響きながら移動するハメになった。

ムロロ（街）にはその日の最終バスで行ったんだけど、終点で「ここには宿泊所はない。」って言われて詰んだ。と思ったら、おっちゃんが部屋を一室貸してくれた。感謝。このソニー・ラディアンさんが実はなかなかの人で、職業を聞いたら「人々に力を与えること」って言った。各村を回って村人の欲している公共設備を聞き出し、会議をさせ、王族と話をつけ、村人にチームを作らせて、政府から援助を引っ張ってきて、プロジェクトを管理してるらしい。あと、グラミン銀行的な農民集団への少額貸付も政府と銀行からお金を引っ張ってきてしてるって。少額貸付のグループは82つ作ったらしい。



最終日の宿泊所では「西スンバは村同士のケンカでいっぱい死人が出て少し危ないから東に引っ越したんだよ」とか従業員が言ってたけど、それって.....。

帰国

スンバ島から抜けてバリに着いて、バリから帰りの航空券を買おうと空港内をうろついてたら航空券のダフ屋に声を掛けられた。とりあえずタバコを勧めていろいろ話を訊いたら、10~15人のグループでしてる、5000円の航空券を6000~9000円で売ってる、一日25枚の航空券を転売してる、売れなくて捨てる分は多くて2、3枚程度、一人当たり一日当たり収入は大きくて400円程度だけどオレは500円稼ぐ、とかいろいろ興味深いことを知れた。チケット転売って資本主義の悪い部分のカスみたいなもんだから、行動経済学的にこういうの調べると面白いかもしれん。



それから何日かバリに居ただけど、手持ちのお金が寂しくなってきた。こんな時のための国際キャッシュカードやで。と思ってUnion PayをATMにつっこんだら6ケタの暗証番号を訊かれて、「おっかしいなあ~」を繰り返したらカードが凍った。背筋も凍った。やばば。

空港使用税とかを計算しつつタピオカ澱粉1kgとかヤシ砂糖・海水塩500gとかを買ってバッグに詰めてジャカルタ空港に行ったら、X線を通したところでセキュリティに「Stop!」って止められた。「Open your bag.」Yes。「What is this powder?」ええと.....this is starch, ini pati.（これは澱粉です）「Why did you buy?」Saya mahasiswa ekonomi pertanian, saya ingin membuat makanan Indonesia di Jepang.（私は農経の学生で、私は日本でインドネシア料理を作りたいです。）「?????????」（インドネシア語で何言ってるかわからん）」みたいなことになって変な汗が。アフリカの主食のウガリを作ってみようと思っただけなのに.....。

緊張からの挙動不審が落ち着いて、空港使用税を払ったら小銭しか残ってなくて、とかなりなが

らイミグレ。パスポートを出す。スラム マラム。しかし、ここでも係員が僕の事をしかめっ面で見つめる……。「お前知ってたか？ このビザが有効なのは今日までで、しかも今は夜中だ。」はい。「もう少し遅かったらお前は出国出来なかった。」はい。「こんなに長い間、インドネシアで一体何をしてたんだ？」ええと、それは……云々。やばい。

で、飛行機乗って夕食逃さないように神経張りながらうたた寝しつつ成田に帰って、ATM 閉まってるしお金ないから PASMO で秋葉原に行って、「掃除しに來い〜。」^{*1}って誘う友人^{*2}と夕食を食べ^{*3}て、京都までの夜行バスに乗ろうとした^{*4}らバスがどこに來てるかわからなくて夜中の東京駅周辺をおっきなバッグ抱えながら探しまわって探しまわって、やっと見つけて^{*5}乗り込んだら直ぐに発車したりした。おなかいたくてインドネシア語で「トイレはありますか？」って寝ぼけながら頭に浮かべつつ京都駅に着いて、トイレに駆け込んだら紙が無かったけど旅行中だからロール持っててよかったね。オワリ。



雑感

インドネシアの掲げる「多様性の中の統一」に上手く乗っかれてるのがタナ・トレジャなのかなーと思ったりした。多様性の中で自分たちの文化を大切にする姿勢は必要になってくるんだと思う。おとなりの権威主義的国家マレーシアでは「一つのマレーシア」が叫ばれてて華僑への“マレー人へ融和せよ”圧力とかを結構感じたんだけど、ここはそんなことなかった。

新自由主義的構造調整政策とイスラム教の喜捨の精神のせいかな、お金の汚い人が多い。ミニバスで降りる際になって乗る前の倍の値段を言われたり、無理矢理ホテル案内してお金せびってきたり。もうちょっと面子とか無いのか。やっぱり、新自由主義は人間の心をな。

グローバリズムで世界が均一化していく中で、自分たちの文化に重点を置き誇りを持つ方針は多様性を維持する観点からも評価されるべきだと思う。多様性は革新の源泉だし、均一化は外からの衝撃に構造的に弱いし。何かしらの価値観に拠る規制で行き過ぎた複雑性を回避しつつ多様性を維持開発していくのが重要になってくるのは、国家であれサークルであれ人類であれ同じ事なのかな、と思いました。はい。

*1 この夏は諸事情で掃除しに行けなかった。ゆるしておくれ。

*2 放って置くとそのまま死んでそうなぐらい生活力がない。

*3 すき家だったんだけど、もっと雰囲気のあるところに行けばよかった。

*4 そのままその友人の家に泊まるべきだったんだ、なにをしてるんだほくは。うろう。

*5 このまま見つからずに彼の家で泊まったらよかったんだ。

とびうおのリア充物語——沖縄編 3

@tobiuo12730

9月

何も進展なしに終わってしまったホームステイが終わり、僕は別の農家さんの家にホームステイしに行きました。8月24日に比嘉さんのライブがあることがわかっていたので、僕はその日を待ち望みにしていました。しかし8月の終わりに台風が来るということが分かり、ガンの手術を受ける祖母が不安を覚えて死にそうな声で電話してきたせいで、8月23日に突然内地に帰ることになりました。

祖母の入院中に要介護者である祖父をある程度介護するのが僕の役目でした。タイミング悪く帰されたせいで僕はこの時期かなりの毒気を放っていたようです。気晴らしに外に出かけると、自分の運の悪さと家に帰ったときの憂いで、数分おきに目眩がするなどの症状もみられました。最初の約束では9月10日に入院が終わり、僕はまた沖縄に戻ることだったのが、祖母の入院が長引きそうということで僕はとても鬱屈としていきました。

そんな様子の僕をさすがに見かねたらしく、神戸の実家に祖父をひきとると進言してくれました。僕の沖縄旅行はただ遊んでいるわけではなく内地での傷心を癒すという大義名分がある、と自分に言い訳をして祖父を母に押し付け、僕は沖縄へと戻りました。

この時期には、沖縄には京大の人たちが宿舎に泊まっていました。彼らと一緒に農作業をしながら、比嘉さんの次のライブを待ち望んでいましたが、次のライブの開催場所が北谷町（東村から100kmむこう）だと聞かされ到底辿りつけるはずもありませんでした。

京大の人たちが帰る日の宴会で、思い切って比嘉さんと呼ぼうと家に電話をかけて見ました。地元の小中学生にも来てよ来てよと言ってもらいましたが、弟さんが来て姉さんのほうは来ませんでした。

京大の人たちが帰り、農家さんのところにホームステイしているとき、地元の小中学校の運動会を比嘉さんも見に来るなら僕と一緒に見ようよと誘ってみたりしましたが、彼女は運動会に現れませんでした。このあたりから、どうも彼女は僕のことを避けてるのだろうなとぼんやりと感じるようになりました。

このようなことがあり、僕はわざわざ沖縄に戻ってきてまで何をしにきたのだろうとしょんぼり

していました。そんなとき、ホームステイ先の農家さんと偶然名護で飲みに行く機会があり、農家さんは僕を名護湾を臨む浜辺に連れて行ってくれました。イカ漁の眩しい光を眺めながら、僕は比嘉さんとの出来事を喋ってみました。すると農家さんはこんなふうに言いました。

「一人の人だけを追いかけるのはやめなさい。もっと多くの女の子とあそんで、いろんな人がいることを勉強しなさい」

あの頃の僕にはまだその言葉が何を意味しているのかはよくわかりませんでした。今ではその意味を徐々に理解してきていると感じています。9月の出来事は僕にとって寂しいことばかりでしたが、この時に感じたことは8月の楽しかった出来事よりも勉強になったと思っています。

バスの中で

「あ、もしもし大輔くん？ 今沖縄にいるならスタバの沖縄限定カード買ってきてよ！」——ツアーの女の子、電話で

「出世したときはうちの娘をお願いします」——ホームステイ先のアルバイトの兄ちゃん

「比嘉さんの父親さんが『男なら女の子にアタックしないとイケない』とか『娘がとびうおさんと付き合うことには特に抵抗しないよ』とかいってはったよ」——京大ツアーの先輩

小学校の運動会が終わった3日後に、東村集落対抗運動会がありました。その行事に参加するには予約が必要なため、僕はただぼーっと眺めていました。小さな村とはいえ東村には僕の知らない人々がたくさんいるのだなど、僕は疎外感を覚えました。そろそろ僕はこの村を出る時期だ、そう肌で感じました。

東村運動会の打ち上げは本来村民しか参加できないはずだったのですが、2ヶ月もいた僕は既に扱いが村民だったらしく、打ち上げの準備をしたかわりに参加させてもらいました。僕はこの地方のしきたりに従って片手に泡盛をもちつつ、村民の方々に挨拶をして回りました。ほとんどの方はエイサーや農業体験で見知った顔で、多くの方は「お前まだいたのか」と驚いていました。また来いよ——そう言ってもらい、さっき感じた疎外感が途端に薄れ、この村に対する愛着のようなものが湧いて来ました。

青年会のタキさん（仮名）「大輔くんどうやって那覇空港まで行くんだ？」

『川田線を使って名護まで行って、そこから高速バスです』

「川田線って比嘉さんの通学路じゃないか」

『実はそこは狙ってました』

タキさんは僕の腕をむんずと掴み、ホームステイ先の奥さんのところまで連れて行きました。こいつ、比嘉さんが通学路に使ってるバスを調べてたんすよ。奥さんは、あら、それは危険だから私が名護まで送ってあげようか、と答えました。僕は、すいません最後に自分の気持ちを伝えたいのでやめてくださいと答えました。タキさんは、よくここで断れるよなあ、呆れた顔でした。翌朝、僕は1日に3本しかないバスのためにバス停で腰を下ろしていました。内地の友達にメールを

やりとりしながら、2ヶ月間眺め続けた風景をもう一度目に焼き付けました。そしてバスがやってきて、僕はそれ乗り込みました。前から2番目の席に、比嘉さんは座っていました。

やあ、今日で僕は内地に帰るんだよ、と軽く会釈をしながら、僕はバスの廊下をはさんで隣側の席に座りました。バスを見渡すと、比嘉さんと同じ制服を着た生徒がちらほら乗っていました。

バスのエンジン音はかなり大きく、廊下をはさんだむこうにいる比嘉さんには大声でないと伝わりません。かといって、今ここで大声を出すと比嘉さんの同級生に噂が立つのではないか——僕は急にそんなことが気になりました。声をかけようかかけまいか——横を見てみると、比嘉さんは携帯電話に見入っていました。

これだ、僕はそう判断して、比嘉さんにメールを送りました。

やあ！ 最後に会えてちょっとうれしいよ！

太平洋側の東村を抜けて日本海側の大宜味村に入っても、比嘉さんから返信は来ません。このままではだめだ、何か別の話題を振らなければ——そうしながら比嘉さんのほうをむくと、窓のむこうに古宇利島が見えました。

いま窓のむこうに見えるのは古宇利島？

バスは日本海沿いを西へ西へと進んでいきます。名護市に入り、海が見えなくなったあたりでようやく返信がきました。

そうですよ（ひよこ）

って、なんでこの距離でメールなんですか（雷）

とっさに

それもそうやな

と返しました。彼女に近づこうとしても、1番前の席には人が座っていて、僕はどうにも近づけません。

名護の市街地に入り、1番前の席の人が降りました。僕はその席にすかさず座り、比嘉さんに軽く会釈をしました。僕の心の中には、告白しなければならぬ、けど告白して断られたら二度と東村には戻れない、そのような気持ちが揺れ動いていました。

行き慣れた名護のジャスコを通りすぎ、名護十字路に入ったあたりで、僕は思い切って比嘉さんに声をかけました。しかし、エンジン音にその声はかき消され、彼女はまったく気づいてくれませんでした。

いつもは1,500kmも離れているのに、今はこの1mが遠い——。バスはそのまま、終点の名護BTに着きました。

僕はバスを降り、入り口付近で荷物を下ろしました。比嘉さんが降りてきた所で、僕は彼女に、またいつか会いましょう、と言いました。彼女は、はい、それじゃあ、と答えて、学校の方向へ歩いて行きました。僕はその後姿をじっと眺めながら、夏の終わりを感じました。

あとがき

とびうおのリア充物語——沖縄編 4 (@tobiuo12730)	54
編集後記 (@d_possum)	55

とびうおのリア充物語——沖縄編 4

@tobiuo12730

まとめ

「東村のタキくん？ ああ、お友達ですよ。昨日あの子と一緒に飲んだの？」——名護 BT のデヴィ夫人『弁当屋のおばちゃん』

僕が比嘉さんに対して抱いていた恋心は、きっと僕が内地で経験したことの反動で生まれた恋心だったのでしょう。きっと9月に入って内地での生傷が癒えたと同時に、彼女に対する心も徐々に収まってしまったのでしょう。バスの中で僕が思いきれなかったのはそれも一因だと思われます。彼女にとってはいい迷惑だったことでしょう——そのことを思うと今も僕は心が痛みます。

『とびうおのリア充物語——沖縄編』、いかがだったでしょうか。読者の皆さんも、このような苦い経験を一度は味わってみるのもよいのではないのでしょうか。最後に、このような機会を与えてくださった東村のみなさん、早稲田大学のみなさん、京都大学のみなさん、さらに内地でいろいろな助言を与えてくださった KMC の皆さんに篤くお礼を申し上げます。

次は読者の皆様とお互いにリア充になってお会いしましょう。それでは。

編集後記

@d_possum

この度は『独習 KMC vol.2』を手にとって頂きありがとうございます。今号編集長に選ばれた先々月がとても遠く感じます。京大マイコンクラブ (KMC) 3 回生の possum です。

前号が情報系研究室の学生と KMC に入会するような初心者に向けたものだったのに対して、本日発行しました vol.2 は対象読者が大きく異なる内容となってしまいました。特に、沖縄に行った彼の記事を全面に押し出しているのは否定できません。前号のような内容を期待されていた紳士淑女の皆様には申し訳ありませんが、今号は前号に対する長めのコーヒープレイク (中には多少例外がありますが。) とお思い頂ければ幸いです。

今回は読者の皆様のお声をお聞きするために読者アンケートを設置いたしました。下記のアドレスで Google Docs のアンケートフォームへリダイレクトされます。

『独習 KMC vol.2』読者アンケート——<http://kmc.jp/c81enq>

皆様のご意見ご感想をお待ちしております。

本誌の制作に当たり、記事執筆者の皆様にはもちろんのこと、その他の皆様にも大きくご協力頂きました。魂を削って表紙を描いてくれた gire 君、本文のアクセントたる挿絵を描いてくださいました Moko さん。文字ばかりの部誌に花を添えて頂きありがとうございました。

vol.1 の編集長であり本誌制作環境を構築してくださった seikichi さん、前々号 vol.β から踏襲している裏表紙のデザインを作り上げた crys 君。お世話になりました。

整形、校正、印刷までの広い範囲で手助けしてくださった花月さん。貴女無しでは本誌は無かったです。ありがとうございました。

拙い文章ではありますが、以上で編集後記とさせていただきます。皆様良いお年を。

平成 23 年 12 月 31 日
とびうお君に早く春が訪れることを祈って。

@d_possum

独習 KMC vol.2

2011 年 12 月 31 日 初版発行

著作・発行 京大マイコンクラブ

表表紙デザイン gire

裏表紙デザイン crys

挿絵 moko

メールアドレス info@kmc.gr.jp

Web <http://www.kmc.gr.jp/>

落丁・乱丁の際は在庫がある限りお取り替えいたします。上記のメールアドレスまでご連絡ください。

独習KMC vol.2

■利用分類

部誌

■製品分類

KMC

■レベル

