ABC120 の感想を SATYSF$_{\mathrm{I}}$ でかいてみる

八釘 かと

1. 概要

今回 A と B を haskell によってときました (C は順次入力していく方式を取りたかったもののわからずに C++ へ，D は ...)．同じ関数型である SATYSF$_{\mathrm{I}}$ を用いて簡単に解き方，感想を書いていきたいです．まだまだ haskell に慣れていないのであまりきれいな書き方ではありませんがご容赦ください．

2. 解き方，感想

2.1. A - Favorite Sound

自販機の音をたくさんききたい ... でもまあ C 回も聞けば満足ぅ ... てことはつまりまあ $min(div\ b\ a, c)$ ということになります (div は商となります)． ソースコードはつぎのように．

- import Control.Monad
- import Control.Applicative
- solve a b c
  - |c <= div b a = c
  - |otherwise = div b a
- main = do
  - [a,b,c] <- map read . words <$> getLine
  - print (solve a b c)

... どうやってプログラムのソースコードいれるのがいいかまだちょっとよくわかりません，すみません．内容としては，do 下一行目で a，b，c を入力内容で束縛し (この表現でいいんでしょうか)， そのあと solve で判定等をし解を print というものとなりますパターンマッチ楽しい！

2.2. B - K-th Common Divisor

公約数問題ですね．ABC ではよく見る気がする (気がする) ね．今回制限が強い ($1 <= A, B <= 100$)

ので，単純に大きい方から A, B を割り切れるかそれでいいはずです．ソースコードはつぎのように．

- import Control.Monad
- import Control.Applicative
- solve i a b 1
  - |(mod a i ==0)&&(mod b i ==0) = i
  - |otherwise = solve (i-1) a b 1
- solve i a b n
  - |(mod a i ==0)&&(mod b i ==0) = solve (i-1) a b (n-1)
  - |otherwise = solve (i-1) a b n
- main = do
  - [a,b,k] <- map read . words <$> getLine
  - print $ solve a a b k

main でやってることは A とそんなにかわりません．solve では，もらった値 i で A, B を割り切れるか判定，割り切れた回数 (つまり大きい公約数) でいままでいくつ見つかったかの保存，i-1 で次の処理というのを順々に繰り返します．目的の k 番目のものがみつかったらそのときの i をかえして print．パターンマ (ry (... ちなみに小さい方からやろうとしてあれ ??? ってなってたのは内緒だよ)

2.3. C - Unification

わからんかったので C++ でときました ... ソースコードははずかしいので内緒．二色あれば絶対に別色のならびがあるので単純にシミュレーション (基本的には来たものを順番にかさね，もし一番上と別色が出たらその一番上と来たやつをとりのぞく) したら解けました．でも解説の 2*$min(0num, 1num)$ のほうが賢い解き方のように思われます．