Advent of Code

Day Eight

LucidBrot

August 2020

1 About

The task at adventofcode 2019 day 8 is fairly straightforward itself. It can be summarized as

> Read the input line of N numeric characters into *layers* of size $width * height$ (which are known) to find the layer that contains the lowest number of zeros. Then return the number of `'1'` digits multiplied by the number of `'2'` digits within that layer.

However, we're doing this in LaTeX, which is typeset in spongebob-case for a reason.

2 The LaTeX Ėẋṗėŕiėńċė

First of all, we're doing something that it was not meant to be used for – so that means we never get the search results we want. Searching about arrays in LaTeXfor example gives you an explanation about how to typeset matrices. Very useful, but not what I wanted. Thankfully, the pgfplots sourceforge page contains a pdf with *Notes On Programming in TeX*.

Secondly, there don't seem to be any variables. Just *counters*, *counts* which are the TeXversion. and *ifdefs* and most importantly *macros*. But I did not read up on the internals of TeXand LaTeX, so I have no clue about the exact way that macros are evaluated. Sometimes you can define a command that works perfectly well for a constant argument, but if you dare use it on the result of another command, you're being had from multiple directions. Because that result has not already been evaluated (expanded) and is passed as-is into the other command. My version of pdfLaTex does not feature the primitive `\expanded` yet. Using `\expandafter` feels very clunky. Luckily there's a hack around that to be found here. And sometimes the problem was actually the `xstring` package which also breaks the hack.

> The macros of this package are not purely expandable, i.e. they cannot be put in the argument of an \edef. Nestling macros is not possible neither.
>
> For this reason, all the macros returning a result (i.e. all excepted the tests) have an optional argument in last position. The syntax is `[ name ]`, where name is the name of the control sequence that will receive the result of the macro: the assignment is made with an \edef which make the result of the macro name purely expandable. Of course, if an optional argument is present, the macro does not display anything.[1]

After eliminating some problems of this sort by storing the result in a new command by virtue of the optional argument, the same problem still appeared because some commands just don't work due to the same issue, even if they are making use of the optional argument to return that in turn (See Figure 1, Figure 2).

```
\def\getchar[#1]#2{%
\StrMid{#1}{#2}{\numexpr #2 + 0\relax}[\mychar]%
\mychar}
```

Figure 1: This command does not like to be used on a non-constant string.

Finally, the performance of the `xstring` package is whack. It takes more than two minutes to figure out the length of a 15'000 character string. The bash command `wc -c inputfile.txt` does that in less than a second.

3 Introduction

We had 34 Strawberries for this year's harvest. Probably not enough. So we are sad now and solve https://adventofcode.com/2019/day/8.

h e
1
6
000001000001220022
hello world 3 3 300
Image Width: 3 Image Height: 2
I want to loop 6 times for the first layer.
The input file contains 18characters.

References

[1] Gonzalo Medina, *Nest StrLen and ifthenelse commands*, `https://tex.stackexchange.com/a/15424/102826`. Accessed 05.08.2020.

```
@@ -143,8 +143,8 @@ \section{Introduction}
143 143           % assign current char
144 144           \def\currentchar{\getchar[\fileline]{\digitctr}}
145 145     Char Char Binks: \currentchar\\
146     - %         % check if zero
147     - %         \IfEq {0}{\currentchar}{
    146 +           % check if zero
    147 +           \IfEq {0}{\currentchar}{
148 148               \advance \currentlayerzerocount 1
149 149               Advanced currentlayerzerocount to \the\currentlayerzerocount
150 150
@@ -157,9 +157,9 @@ \section{Introduction}
157 157                 \digictr={\the\numexpr \layersize * \currentlayer + \layersize}
158 158           Layer \the\currentlayer has more zeros than the current best layer (\the\bestlayer ) so we skip ahead to character at index \digictr to start the next layer.
159 159               \fi
160     - %             }{%else
    160 +               }{%else
161 161                   The current char \currentchar~does not equal 0. It is \meaning\currentchar whereas 0 is \meaning0.
162     - %         }%fi
    162 +           }%fi
163 163           \ifnum \digitctr<\inteval{\layersize * \currentlayer + \layersize}
164 164           \repeat
165 165           % if there were very little zeros, we can update the best layer
```

Figure 2: The difference between wrong code that compiles (red) and seemingly correct code that produces a compiler error (green).