

PEP Scala (4)

Email: christian.urban at kcl.ac.uk

Slides & Code: KEATS

<https://pollev.com/cftutoratki576>

Scala Install Clinic: This evening at 17:00 (online)

Hints in CW

Hints

For Preliminary Part: useful operations involving regular expressions:

```
reg.findAllIn(s).toList
```

finds all substrings in `s` according to a regular regular expression `reg`; useful list operations: `.distinct` removing duplicates from a list, `.count` counts the number of elements in a list that satisfy some condition, `.toMap` transfers a list of pairs into a Map, `.sum` adds up a list of integers, `.max` calculates the maximum of a list.

For Core Part: use `.split(",").toList` for splitting strings according to commas (similarly `\n`), `.getOrElse(...)` allows to query a Map, but also gives a default value if the Map is not defined, a Map can be 'updated' by using `+`, `.contains` and `.filter` can test whether an element is included in a list, and respectively filter out elements in a list, `.sortBy(_._2)` sorts a list of pairs according to the second elements in the pairs—the sorting is done from smallest to highest, `.take(n)` for taking some elements in a list (takes fewer if the list contains less than `n` elements).

Scala Library, e.g. span in

<https://www.scala-lang.org/api/current/scala/collection/>

Discussion Forum



Re: Core 6 - Getting a little off the target numbers for Part7

by Christian Urban - Saturday, 23 November 2019, 1:06 AM

Hi,

It is a subtle problem, but unfortunately Scala calculates different results according to when you round numbers. As a result `yearly_yield` needs to be careful when numbers are rounded to Longs. For example, if your balance is \$100 and your calculated profit is negative, say -20.5, then

```
100 + ((-20.5).toLong) = 80
```

while

```
(100 + (-20.5)).toLong = 79
```

Hope this helps,
Christian

Last Week: Pattern Matching

```
def fizz_buzz(n: Int) : String =  
  (n % 3, n % 5) match {  
    case (0, 0) => "fizz buzz"  
    case (0, _) => "fizz"  
    case (_, 0) => "buzz"  
    case _ => n.toString  
  }
```

Reverse Polish Notation

$$(3 + 1) * (2 + 9)$$

\Rightarrow

$$3 \ 1 \ + \ 2 \ 9 \ + \ *$$

Reverse Polish Notation

$(3 + 1) * (2 + 9)$

\Rightarrow

3 1 + 2 9 + *

ldc 3

ldc 1

iadd

ldc 2

ldc 9

iadd

imul

Regular Expressions

Suppose you have the regular expression $(a^*)b$:

”aaaaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have the regular expression $(a^*)b$:

”aaaaa.....aaaaaaaaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have the regular expression $(a^*)b$:

”aaaaa.....aaaaaaaaaaaaaaaaaaaaaa”

Regular Expressions

Suppose you have the regular expression $(a^*)^*b$:

”aaaaa.....aaaaaaaaaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have the regular expression $(a^*)^*b$:

”aaaaa.....aaaaaaaaaaaaaaaaaaaaaa”

Regular Expressions

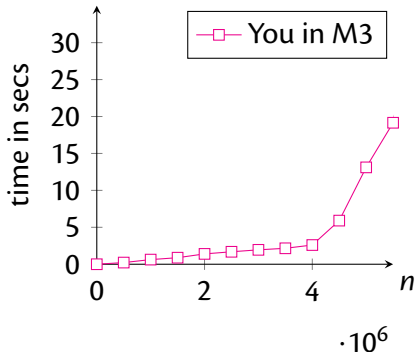
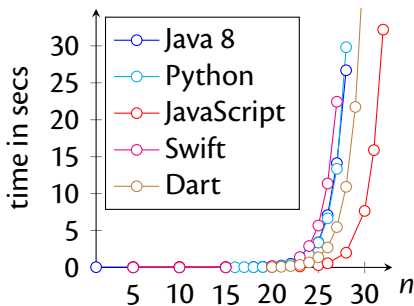
Suppose you have the regular expression $(a^*)^*b$:

”aaaaa.....aaaaaaaaaaaaaaaaaaaaa”

How long does Python need to find out?

Main 3: Regexes

Graphs: regex $(a^*)^*b$ and strings $\underbrace{a \dots a}_n$



<https://vimeo.com/112065252>

MacOSX

- 0) (if needed) `brew install java` or `brew reinstall java`
- 1) `curl -s "https://get.sdkman.io" | bash`
- 2) `sdk install scala 2.13.7`

Windows / Linux Ubuntu

- 0) (if needed) `sudo apt-get remove scala-library scala`
- 1) `sudo wget https://downloads.lightbend.com/scala/2.13.7/scala-2.13.7.deb`
- 2) `sudo dpkg -i scala-2.13.7.deb`

other Linux distros: `sudo apt-get scala`