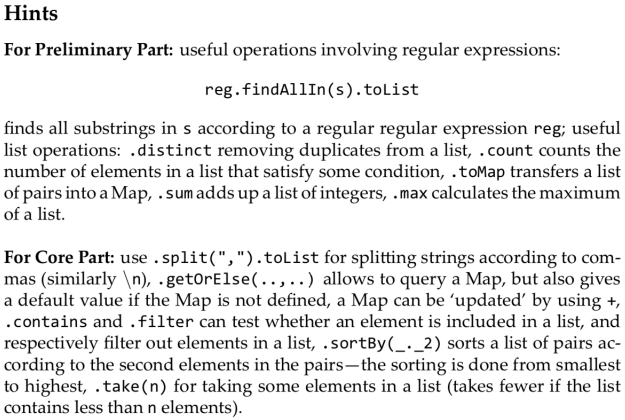
PEP Scala (4)

Email:	christian.urban at kcl.ac.uk
Office:	N7.07 (North Wing, Bush House)
Slides & Code:	KEATS
Office Hours:	Thursdays 12:00 – 14:00
Additionally:	(for Scala) Tuesdays 10:45 – 11:45

PEP Scala (4)

Email:	christian.urban at kcl.ac.uk
Office:	N7.07 (North Wing, Bush House)
Slides & Code:	KEATS
	A Crash-Course in Scala
Office Hours:	Thursdays 12:00 – 14:00
Additionally:	(for Scala) Tuesdays 10:45 – 11:45

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/immutable/List.html`

Preliminary 7

Raw marks (261 submissions):

4%:	236
3%:	10
2%:	1
1%:	0
0%:	15

(interviews ongoing!)

```
def is_legal(dim: Int, p: Path, x: Pos) = {
  if (...some_really_long_condition...) false
  else true
}
```

```scala
def is_legal(dim: Int, p: Path, x: Pos) = {
  if (...some_really_long_condition...) false
  else true
}
```

```scala
def is_legal(dim: Int, p: Path, x: Pos) =
  !(...some_really_long_condition...)
```

```scala
def foobar(...) = {
  val cs = for (c <- str) yield c.toLowerCase
  ...
}
```

```
def foobar(...) = {
  val cs = for (c <- str) yield c.toLowerCase
  ...
}
```

```
def foobar(...) = {
  val cs = str.map(_.toLowerCase)
  ...
}
```

```
def RomanNumeral2Int(rs: RomanNumeral): Int =
 rs match {
   case Nil => 0
   case M::r    => 1000 + RomanNumeral2Int(r)
   case C::M::r => 900 + RomanNumeral2Int(r)
   case D::r    => 500 + RomanNumeral2Int(r)
   case C::D::r => 400 + RomanNumeral2Int(r)
   case C::r    => 100 + RomanNumeral2Int(r)
   case X::C::r => 90 + RomanNumeral2Int(r)
   case L::r    => 50 + RomanNumeral2Int(r)
   case X::L::r => 40 + RomanNumeral2Int(r)
   case X::r    => 10 + RomanNumeral2Int(r)
   case I::X::r => 9 + RomanNumeral2Int(r)
   case V::r    => 5 + RomanNumeral2Int(r)
   case I::V::r => 4 + RomanNumeral2Int(r)
   case I::r    => 1 + RomanNumeral2Int(r)
 }
```

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

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

$$\Rightarrow$$

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

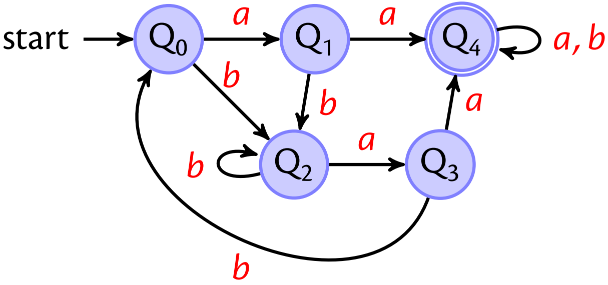
```
ldc 3
ldc 1
iadd
ldc 2
ldc 9
iadd
imul
```

Sudoku

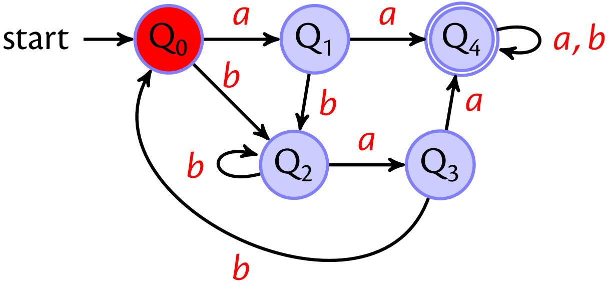
A very simple-minded version on 110 problems:

- 1 core: 800 secs
- 2 cores: 400 secs
- 8 cores: 290 secs
- 18 cores: 142 secs

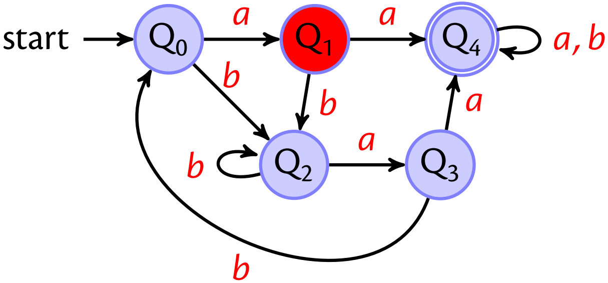
DFAs



DFAs

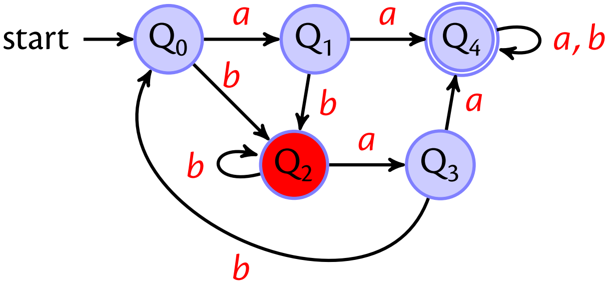


DFAs



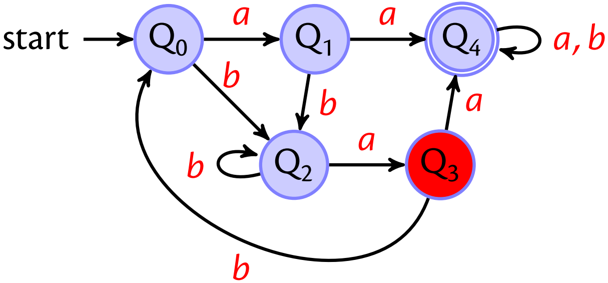
a

DFAs



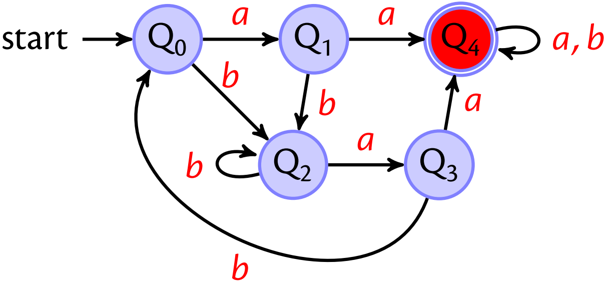
ab

DFAs



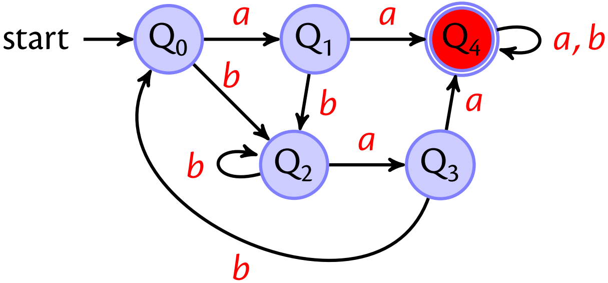
aba

DFAs



abaa

DFAs



abaaa ⇒ **yes**

DFAs

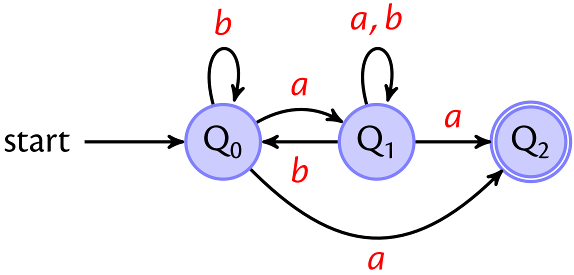
A **deterministic finite automaton**, DFA, consists of 5 things:

- an alphabet Σ
- a set of states Qs
- one of these states is the start state Q_0
- some states are accepting states F, and
- there is transition function δ

which takes a state and a character as arguments and produces a new state; this function might not be everywhere defined

$$A(\Sigma, Qs, Q_0, F, \delta)$$

NFAs



Mind-Blowing Regular Expressions:
in Python, JavaScript, Java

Regular Expressions

Suppose you have a regular expression `(a*)b`:

”aaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have a regular expression `(a*)b`:

”aaaaa...........aaaaaaaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have a regular expression `(a*)b`:

”aaaaa...........aaaaaaaaaaaaaaaaaaaa”

Regular Expressions

Suppose you have a regular expression `(a*)*b` :

”aaaaa. . . . . .aaaaaaaaaaaaaaaaaaaab”

Regular Expressions

Suppose you have a regular expression `(a*)*b` :

”aaaaa. . . . . .aaaaaaaaaaaaaaaaaaaa”

Regular Expressions

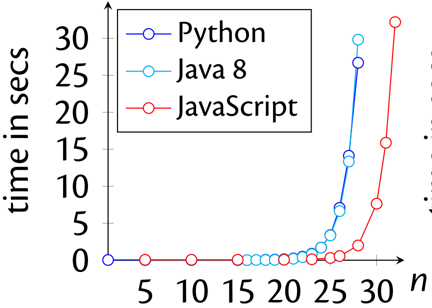
Suppose you have a regular expression `(a*)*b`:

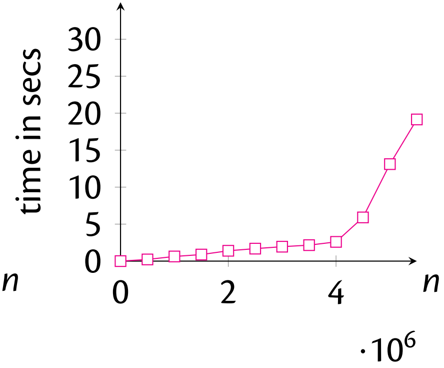
”aaaaa. . . . . .aaaaaaaaaaaaaaaaaaaa”

How long does Python need to find out?

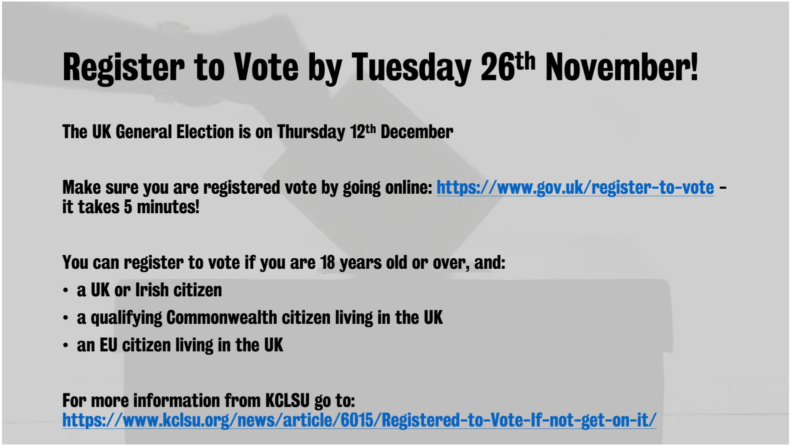
CW 9: Regexes

Graphs: `(a*)*b` and strings $\underbrace{a \ldots a}_{n}$





https://vimeo.com/112065252



Register to Vote by Tuesday 26th November!

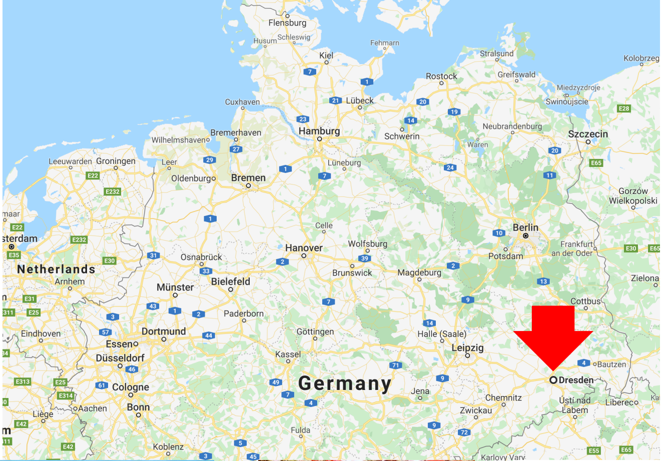
The UK General Election is on Thursday 12th December

Make sure you are registered vote by going online: https://www.gov.uk/register-to-vote – it takes 5 minutes!

You can register to vote if you are 18 years old or over, and:

- **a UK or Irish citizen**
- **a qualifying Commonwealth citizen living in the UK**
- **an EU citizen living in the UK**

For more information from KCLSU go to:
https://www.kclsu.org/news/article/6015/Registered-to-Vote-If-not-get-on-it/


Flensburg
Husum
Schleswig
Kiel
Fehmarn
Stralsund
Kolobrzeg
Rostock
Greifswald
Miedzyzdroje
Swinoujscie
Cuxhaven
Lübeck
Bremerhaven
Wilhelmshaven
Hamburg
Schwerin
Neubrandenburg
Waren
Szczecin
Leeuwarden
Groningen
Leer
Oldenburg
Bremen
Lüneburg
Gorzów Wielkopolski
Celle
Berlin
Potsdam
Frankfurt an der Oder
Amsterdam
Netherlands
Arnhem
Osnabrück
Hanover
Wolfsburg
Brunswick
Magdeburg
Zielona
Münster
Bielefeld
Cottbus
Paderborn
Eindhoven
Dortmund
Essen
Düsseldorf
Göttingen
Halle (Saale)
Leipzig
Bautzen
Kassel
Dresden
Cologne
Germany
Jena
Chemnitz
Ústí nad Labem
Liberec
Aachen
Bonn
Zwickau
Liège
Fulda
Koblenz
Karlovy Vary
Frankfurt

ГОСПОДИ! ПОМОГИ МНЕ ВЫЖИТЬ
СРЕДИ ЭТОЙ СМЕРТНОЙ ЛЮБВИ
MEIN GOTT, HILF MIR, DIESE TÖDLICHE LIEBE ZU ÜBERLEBEN

East Germany
BERLIN
East
West
Berlin Wall in 1961