For SSBSE 2015 Challenge Track:

Regression Test Case Prioritization for Guava

Yi Bian, Serkan Kirbas, Mark Harman, Yue Jia, Zheng Li

Reported by:

Yi Bian

05.09.2015

Authors

Guava

Yi Bian*

Zheng Li*

Serkan Kirbas+

Mark Harman#

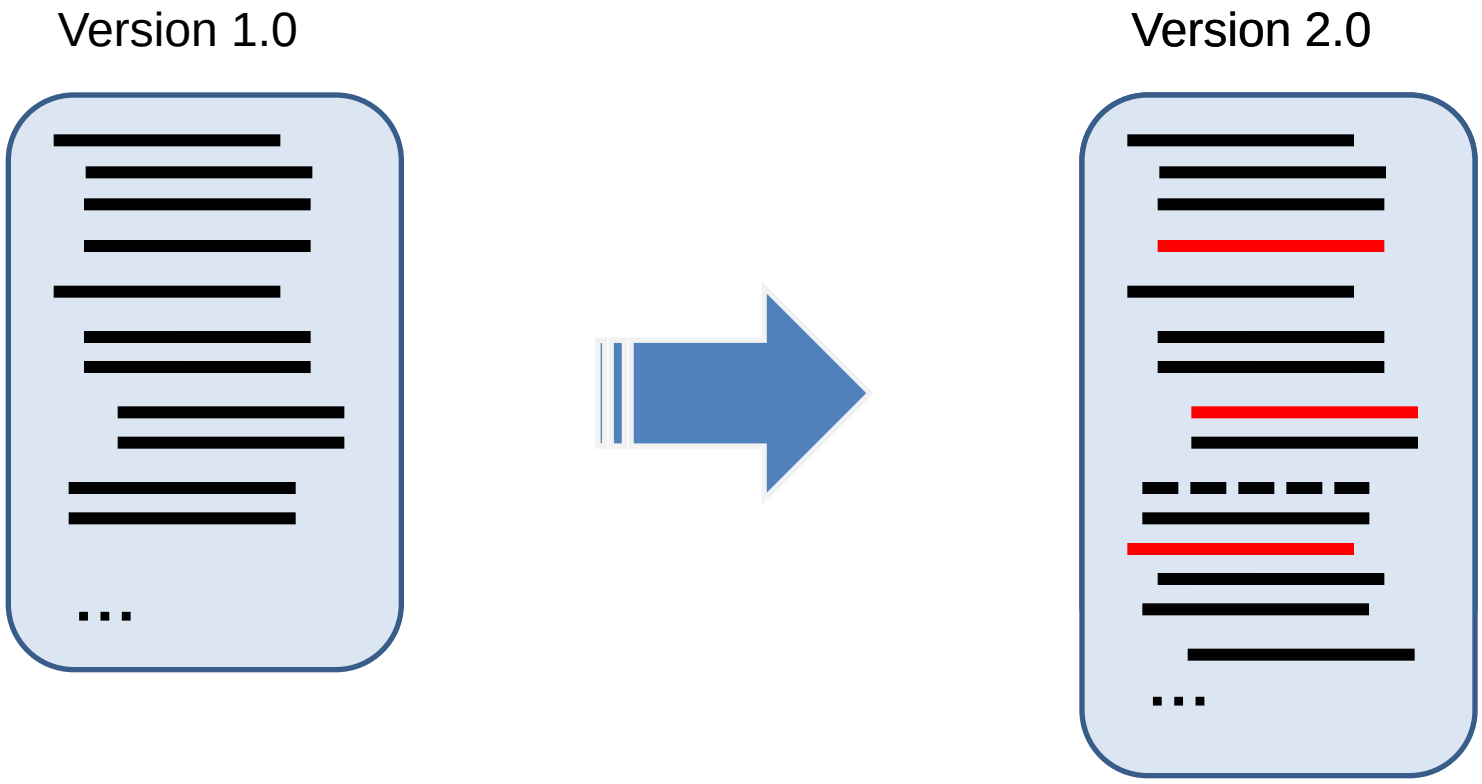
Yue Jia#

University College London, UK

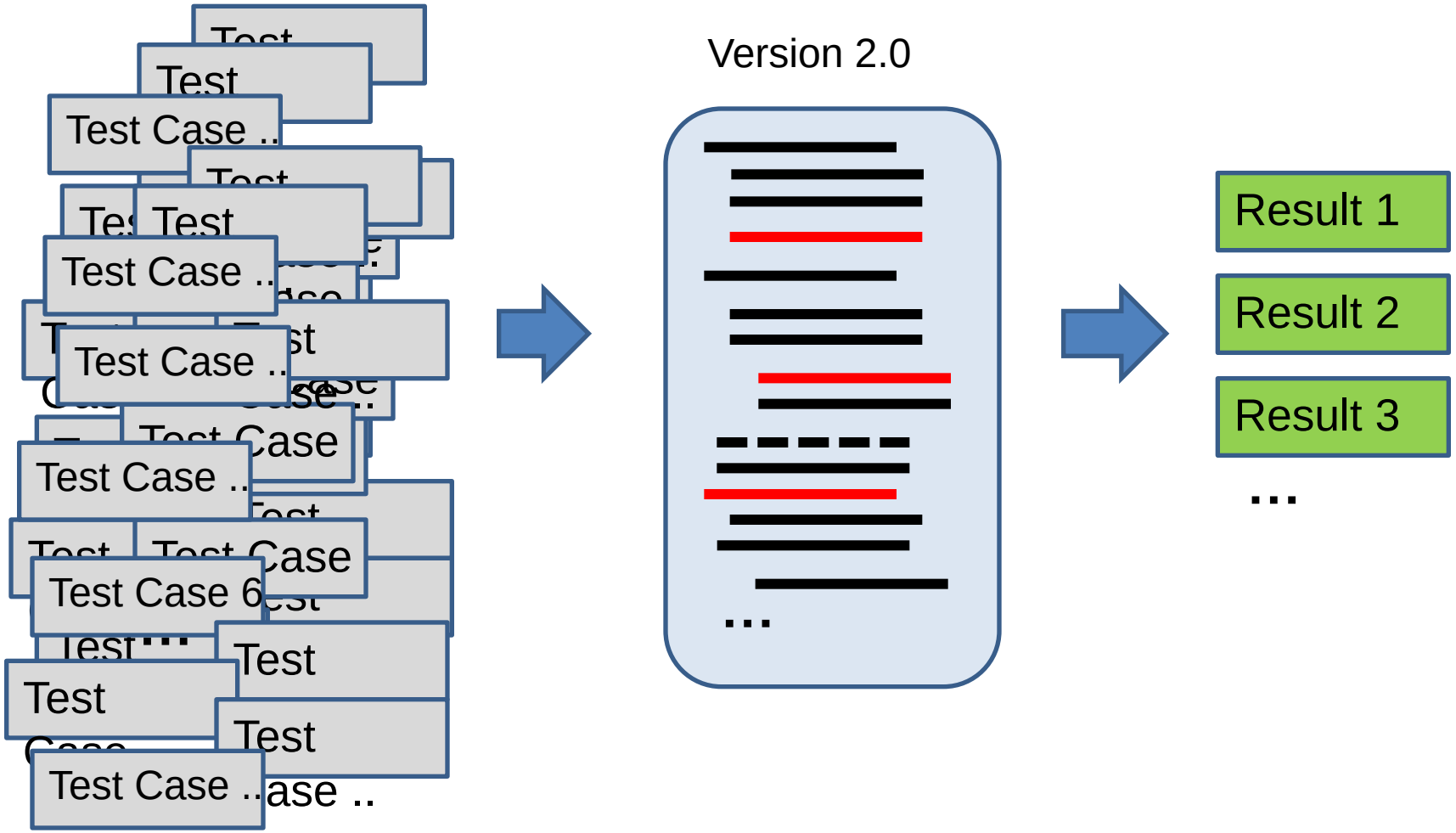
+ Brunel University, London, UK
and Bogazici University, Istanbul, Turkey

* Beijing University of Chemical Technology, China.

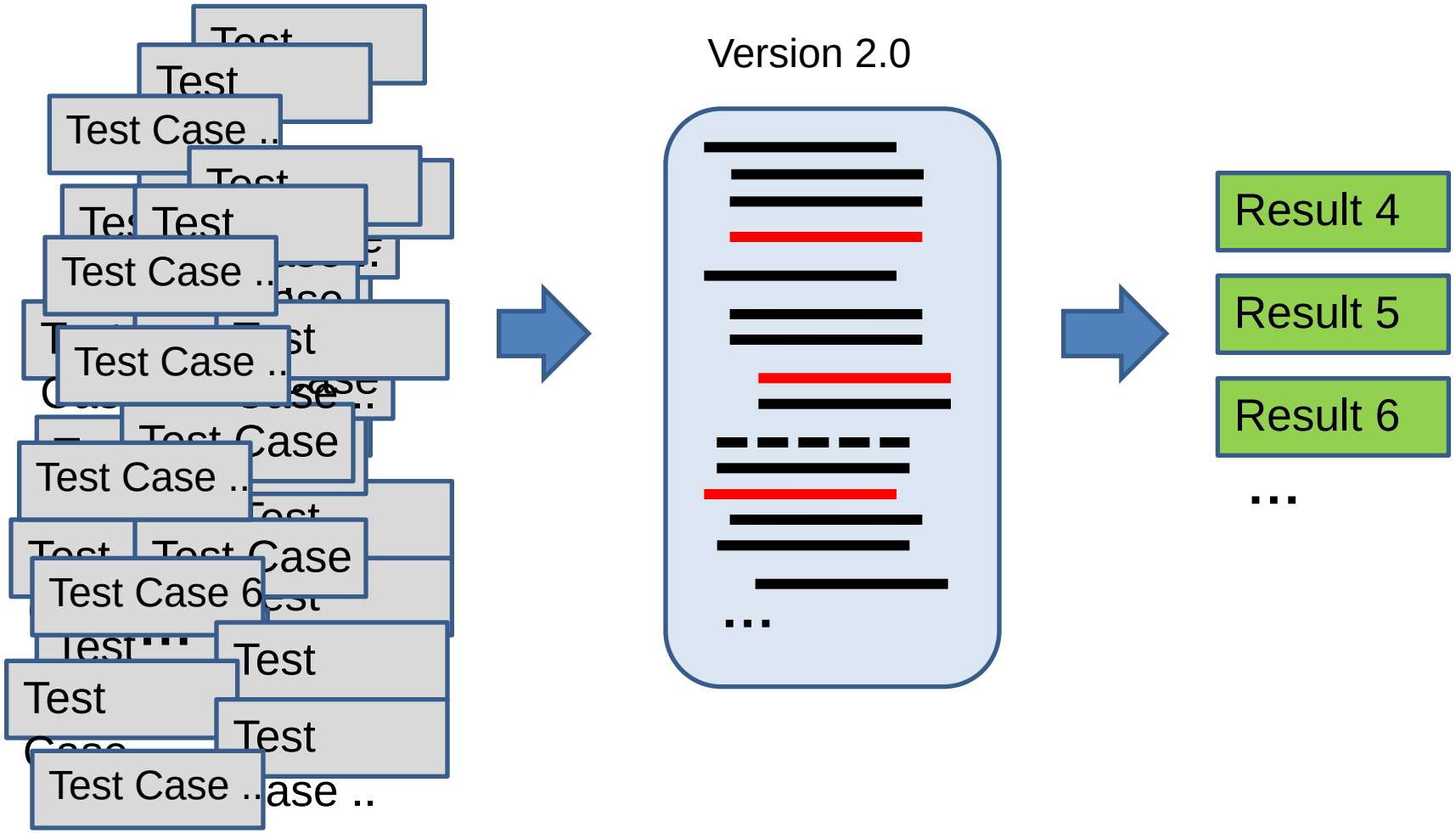
Regression Testing



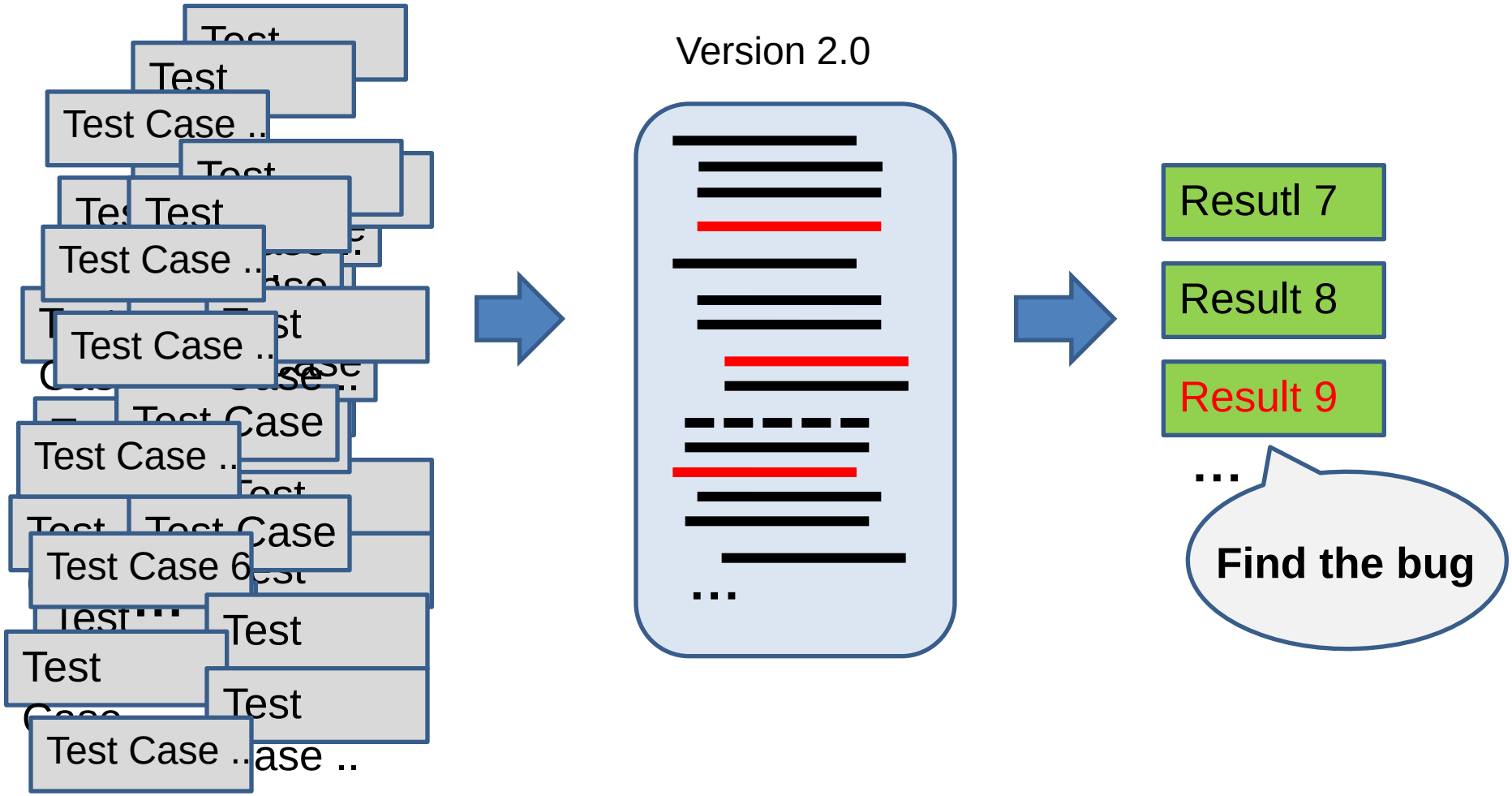
Regression Testing



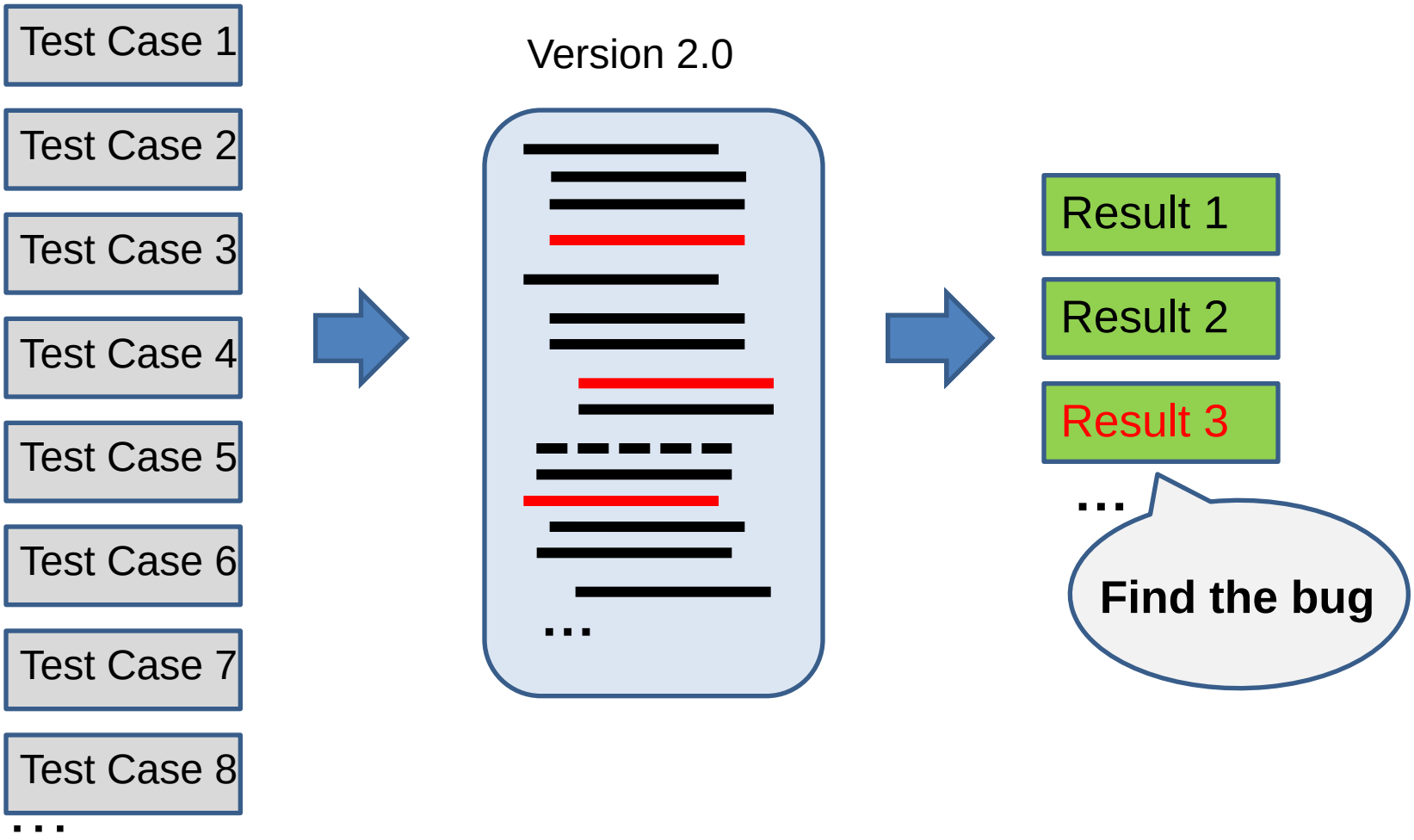
Regression Testing



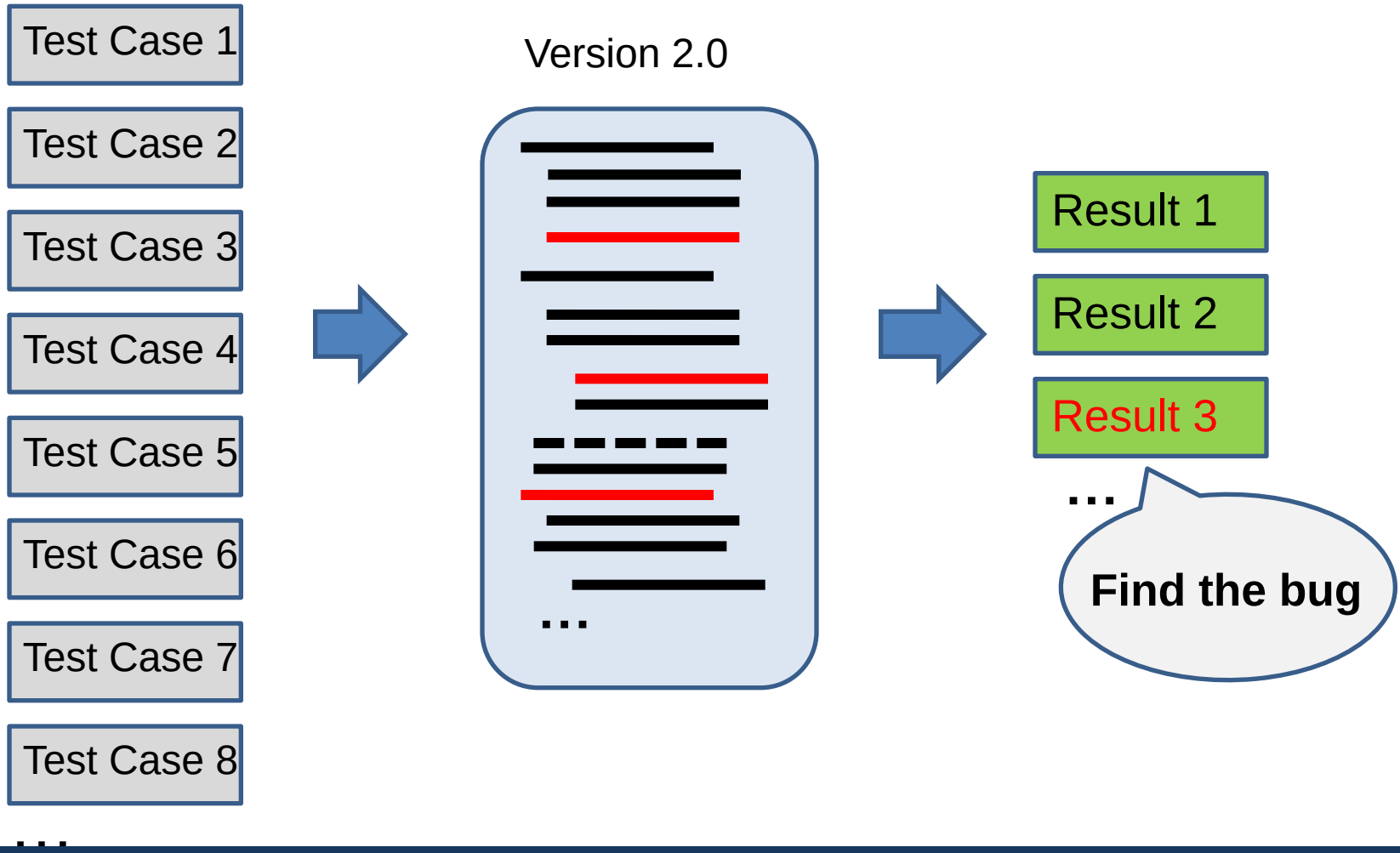
Regression Testing



Test Case Prioritization



Test Case Prioritization



Information that we need

Collection the information for the Guava

Source code:

```
L 1   public static void main(String[] args)
L 2   {
...       try {
                  URL mySite = new URL(“..");
                  …
          }
          catch (Exception e)
          {
                e.printStackTrace();
          }
      }
```

Test Case Sequence:

Information that we need

Collection the information for the Guava

Source code:

```
L 1   public static void main(String[] args)
L 2   {
...       try {
                  URL mySite = new URL(“..");
                  …
          }
          catch (Exception e)
          {
                e.printStackTrace();
          }
      }
```

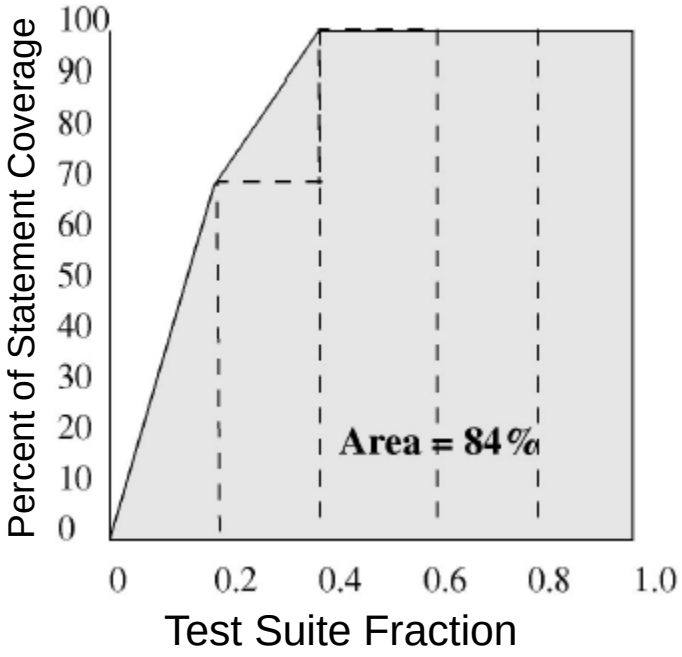
Coverage matrix:

tc1	*tc2*	*tc3*		*tcn*
1	0	0		0
1	0	0		0
0	1	0		0
1	1	0		0
...	...	...		...
1	1	1		1
0	0	0		0
0	1	0		1
0	0	1		0
0	1	1		0
1	1	1		1
Execution time:	0.2s	0.3s	0.5s	0.8s

Prioritization targets

Test Case Prioritization process

Test case sequence: 2 – 4 – 3 – 7 – 11 – 18 – 6 ….



Average Percentage of Statement Coverage:

$$APSC = \left(1 - \frac{TS_1 + TS_2 + \cdots + TS_N}{NM} + \frac{1}{2N}\right) * 100\%$$

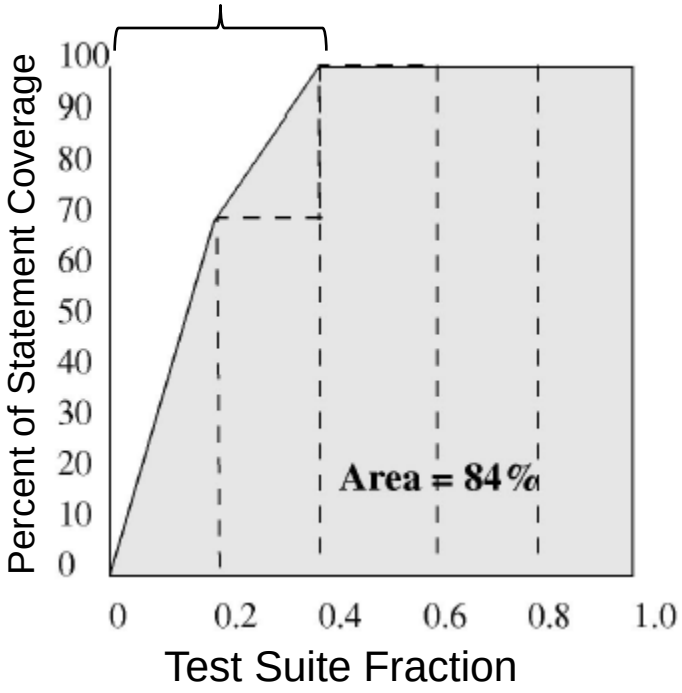
N is the number of test cases,
M is total number of changed statements,
TSi denotes the identifier of the test case that first covers the statement i in the execution sequence.

It is based on the hole coverage matrix.

Prioritization targets

Test Case Prioritization process

Test case sequence: 2 – 4 – 3 – 7 – 11 – 18 – 6 ….



Effective execution time:

$$EET = \sum_{i=1}^{n_{cover}} ET_i$$

n_{cover} is the number of test cases that achieve the 100% of coverage.
ET_i is the execution time of test case i.

It is based on a subsequence.

A new objective

Collection the information for the Guava

Source code:

```
public static void main(String[] args)
{
    try {
            URL mySite = webURL;
            …
    }
    catch (Exception e)
    {
        e.prinURL();
    }
}
```

Change lines:

0
0
0
1
...
0
0
0
1
0
0

A new objective

Collection the information for the Guava

Change lines:	***Coverage matrix:***			
	TC1	*TC2*	*TC3*	*TCn*
0	1	0	0	0
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
...	...	...	...	...
0	1	1	1	1
0	0	0	0	0
0	0	1	0	1
1	0	0	1	0
0	0	1	1	0
0	1	1	1	1

A new objective

Collection the information for the Guava

Change lines:

Coverage matrix:

	TC1	*TC2*	*TC3*		*TCn*
0	1	0	0		0
0	1	0	0		0
0	0	1	0		0
1	0	1	0		0
…	…	…	…		…
0	1	1	1		1
0	0	0	0		0
0	0	1	0		1
1	0	0	1		0
0	0	1	1		0
0	1	1	1		1

Sub coverage matrix:

TC2	TC5	TC7	TCn’
0	0	0	0
1	1	1	0
…	…	…	…
1	0	1	0
0	0	1	0
1	1	1	1

Prioritization targets

Test Case Prioritization process

Average Percentage of Change Coverage:

$$APCC = (1 - \frac{TC_1 + TC_2 + \cdots + TC_{M'}}{M'N} + \frac{1}{2N}) * 100\%$$

N is the number of test cases.
M' is number of changed statements.
TCi denotes the identifier of the test case that first covers the change statement i in the execution sequence.

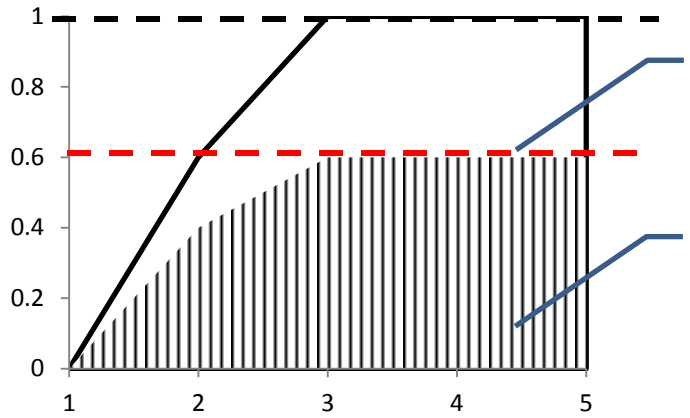
It is evaluate the effective of test case sequence in covering the change lines based on the sub coverage matrix.

Why we need the change lines?

Test case sequence: 2 – 4 – 3 – 7 – 11 – 18 – 6 ….

100% of total line coverage

100% of change line coverage



Area of APSC

Area of APCC

TC 2: (L1, L2, L3) TC 4: (L4, L5, L6)

For APSC there is no different to choose which of them.

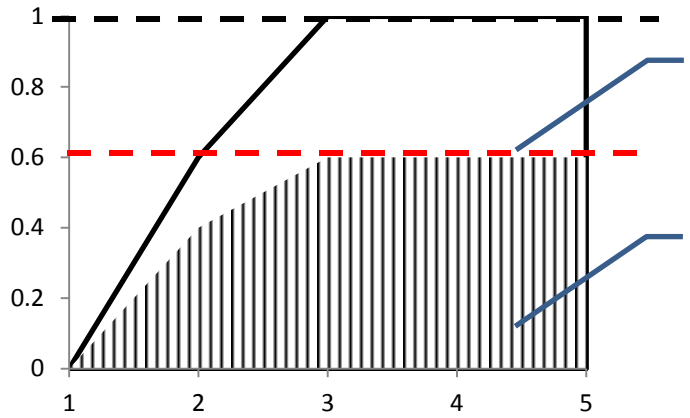
For APCC the TC 4 is better than TC 2.

Why we need the change lines?

Test case sequence: 2 – 4 – 3 – 7 – 11 – 18 – 6 ….

100% of total line coverage

100% of change line coverage



Area of APSC

Area of APCC

APSC is to evaluate effective of test case for total statement coverage.

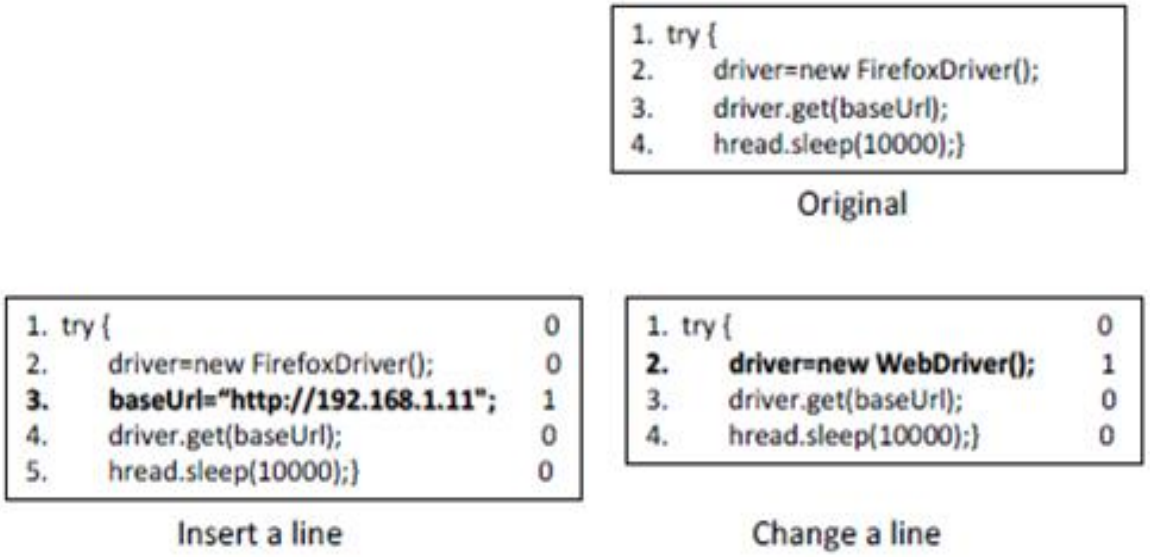
APCC is just evaluate the change line coverage for test case sequence.

There are no different between different lines in programs.

Hope to cover the change lines first.

Change lines

Get the change lines



To extract the code change information, we first extract the current version of Guava from the git repository, and then use the "**Blame**" function to determine which lines have been changed.

Experiment

Different groups:

Table 1. Three groups of experiments are conducted

Group	Optimization Objectives
G1	APSC and EET
G2	APCC and EET
G3	APSC, APCC and EET

Code of guava:

1. In each experiment, there are 26,815 test cases in the optimization sequences for 62 classes in the collection package. Based on this Git analysis, we found that 140 lines have changed.
2. Each with a generation upper limit of 1000 generations. We also terminate the search if the sum value of average change in different optimization objectives is smaller than 0.0001 in 10 consecutive generations.

Experiment

Algorithm: NSGA-II

Table 2: Average value of objectives in different group of experiments

Strategy		APSC	APCC	EET(s)	Length	Time(s)	Generation	Front set
original		24.8085%	9.1243%	210.47	26808.00	-	-	-
random		86.1261%	89.7498%	188.26	24031.36	442.02	-	19.30
100	G1	99.9584%	-	1.13	80.65	71.82	800.09	2.39
	G2	-	99.9834%	0.39	22.94	4.06	82.40	1.24
	G3	99.9563%	99.9900%	1.40	111.32	71.48	721.30	4.84
200	G1	99.9674%	-	1.03	59.02	99.65	477.70	2.78
	G2	-	99.9925%	0.28	7.13	5.47	34.57	1.66
	G3	99.9657%	99.9921%	1.08	65.76	107.39	432.58	5.61
500	G1	99.9709%	-	0.96	49.93	180.33	214.70	2.73
	G2	-	99.9927%	0.28	7.00	12.91	31.50	1.77
	G3	99.9692%	99.9923%	0.94	48.15	210.27	223.69	7.43

random: We generated 100,000 random sequences, repeated this process 100 times.

Length : The number of selected tests to achieve maximum coverage.

Time : Execution time on average. (For random Time is the average time of elite selection.)

Experiment

About the different sequence

In order to avoid double counting, we combined the test cases with the same statement coverage which reduced the number of test cases considered from 26,815 to 14,516 different test cases.

		100			200			500			Total
		G1	G2	G3	G1	G2	G3	G1	G2	G3	
100	G1	-	430	2060	624	71	2221	489	64	571	2591
	G2		-	1008	295	59	1101	237	52	265	1260
	G3			-	1223	140	8889	823	134	945	10758
200	G1				-	65	1337	420	59	483	1486
	G2					-	142	63	30	66	167
	G3						-	859	142	995	11933
500	G1							-	49	448	935
	G2								-	49	165
	G3									-	1066

Total: The total number of test cases in this group of experiments.

Conclusion

For test case prioritization

Test case prioritization is effective for the Guava program.

Multi objective

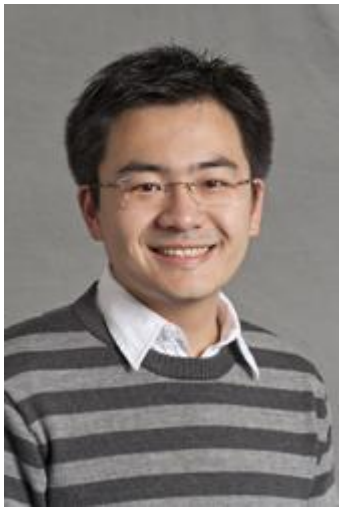
For different test objectives, prioritization sequences are different.

For programs

Considering to have two sprint prioritization processes based on different objectives, the sequences may very different.

Acknowledge

Thanks for supply from other authors !



Serkan Kirbas

Mark Harman

Yue Jia

Zheng Li

Thanks!

Q & A