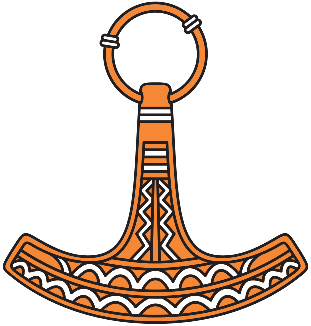
DYALOG

Elsinore 2023

Future Performance Improvements in Set Functions



Karta S. Kooner

October 13, 2023

Assumption that 18.0 was the most performant version

Assumption that 18.0 was the most performant version

But introduced some significant bugs...

Assumption that 18.0 was the most performant version

But introduced some significant bugs...

Version 18.2 removed both the bugs but also the performance enhancements

Reintroduce 18.0 changes back into the interpreter

Reintroduce 18.0 changes back into the interpreter

Require careful analysis and audit of those changes

Reintroduce 18.0 changes back into the interpreter

Require careful analysis and audit of those changes

Presented the start of that analysis last time

DYALOG

Set functions:

- ∊ - membership
- ⍳ - index of
- ∪ - unique, union
- ∩ - intersection
- ~ - without

Set functions:

- ∊ - membership
- ⍳ - index of
- ∪ - unique, union
- ∩ - intersection
- ~ - without

Instrumented:

- ⍸ - where, interval index
- ≠ - unique mask

Set functions:

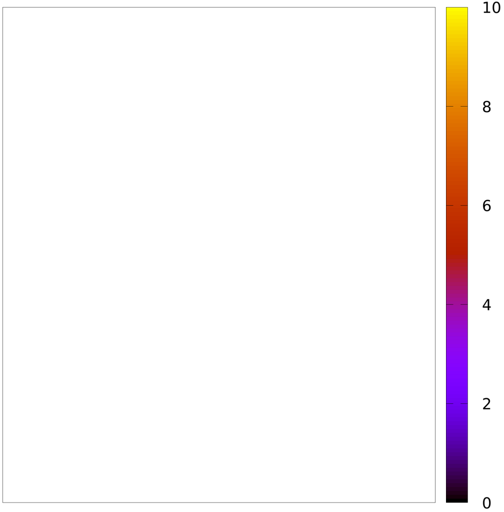
- ∊ - membership
- ⍳ - index of
- ∪ - unique, union
- ∩ - intersection
- ~ - without

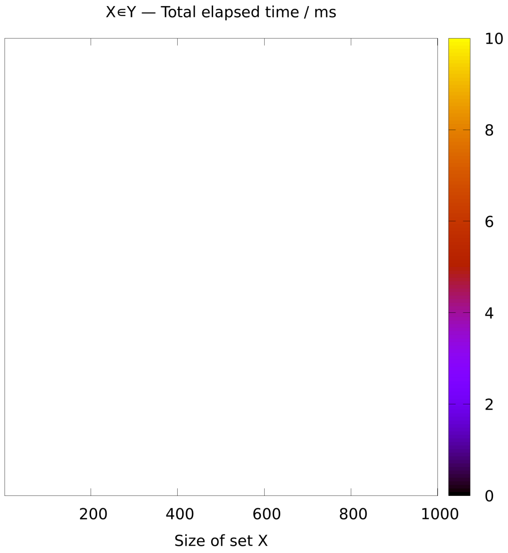
Instrumented:

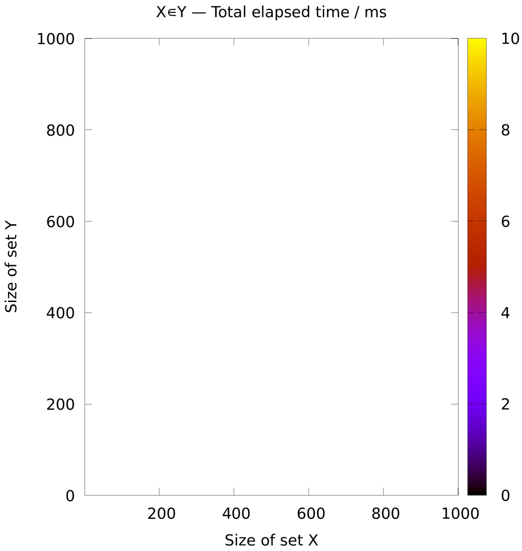
- ⍸ - where, interval index
- ≠ - unique mask
- ⍷ - find

(pause to warn audience)

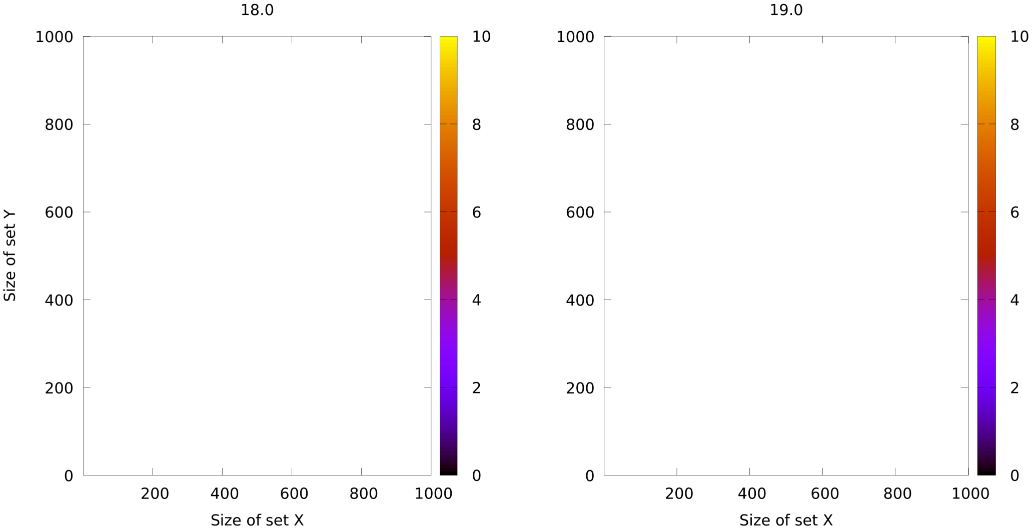
X∈Y — Total elapsed time / ms




X∈Y — Total elapsed time / ms
200
400
600
800
1000
Size of set X
10
8
6
4
2
0

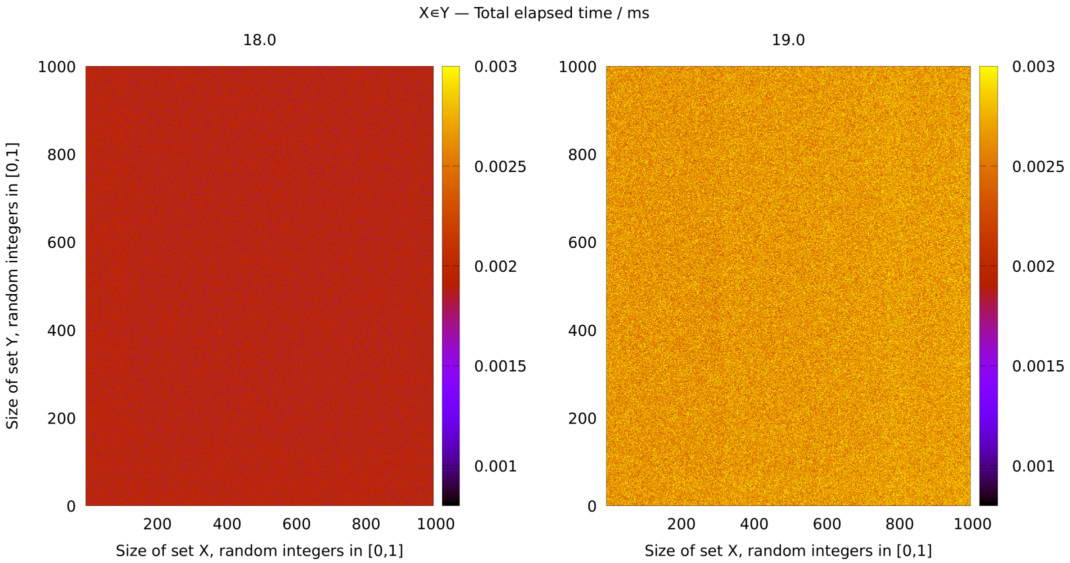

X∈Y — Total elapsed time / ms
Size of set Y
Size of set X
0
200
400
600
800
1000
200
400
600
800
1000
0
2
4
6
8
10

X∈Y — Total elapsed time / ms



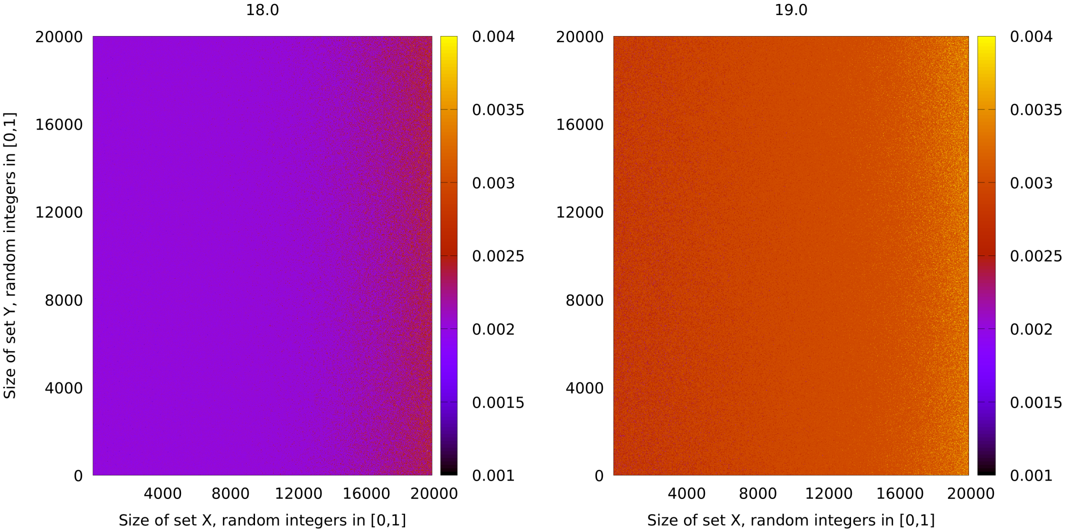
18.0

19.0

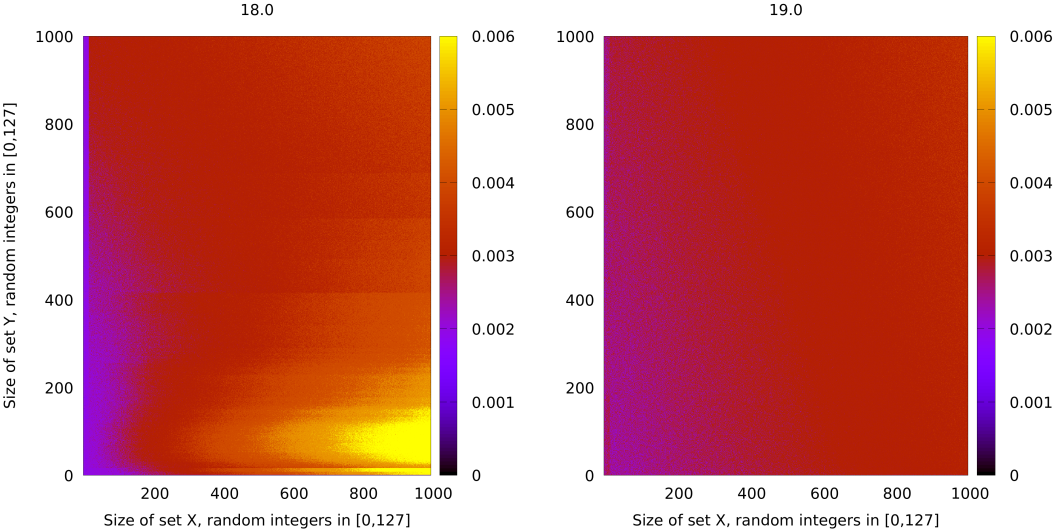


X∈Y — Total elapsed time / ms

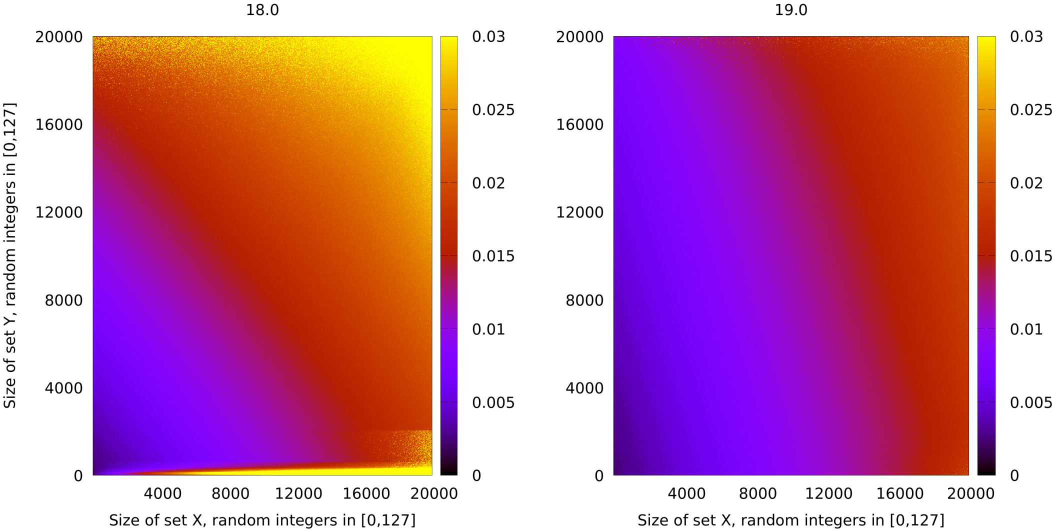
X∈Y — Total elapsed time / ms



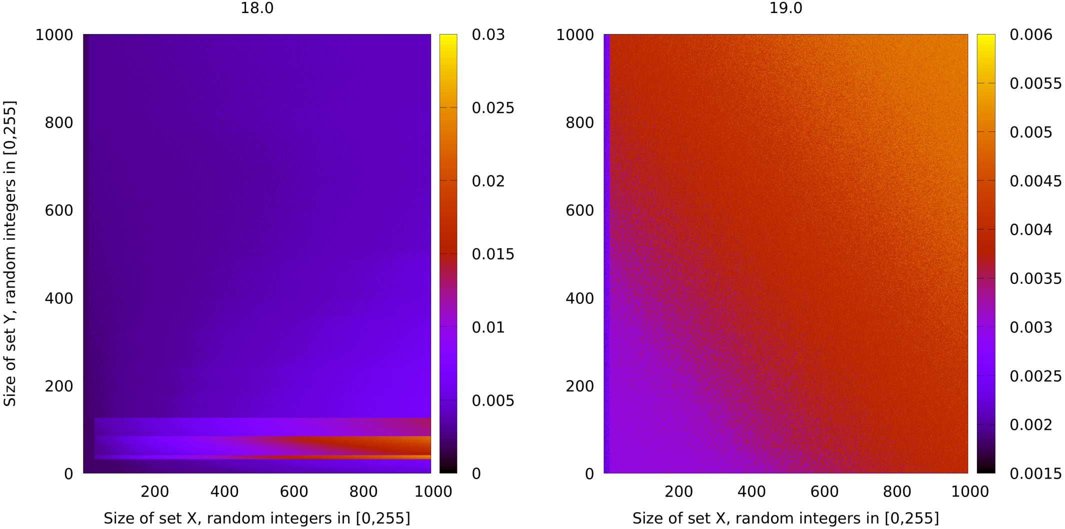
X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



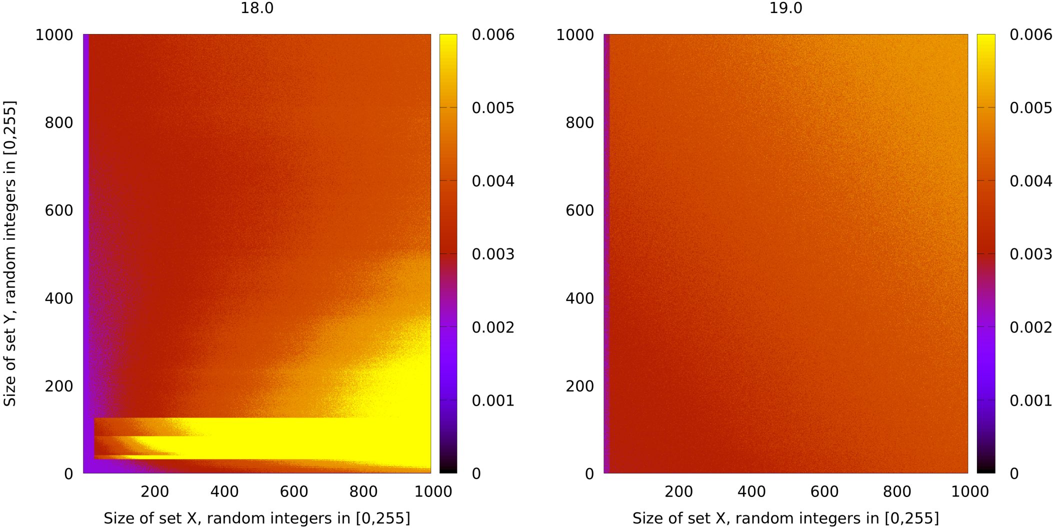
X∈Y — Total elapsed time / ms



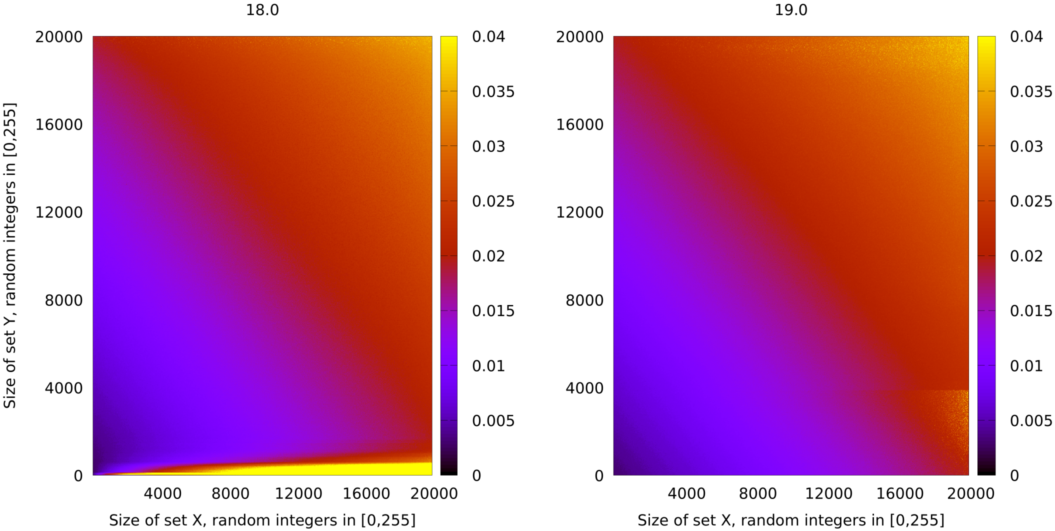
18.0

19.0

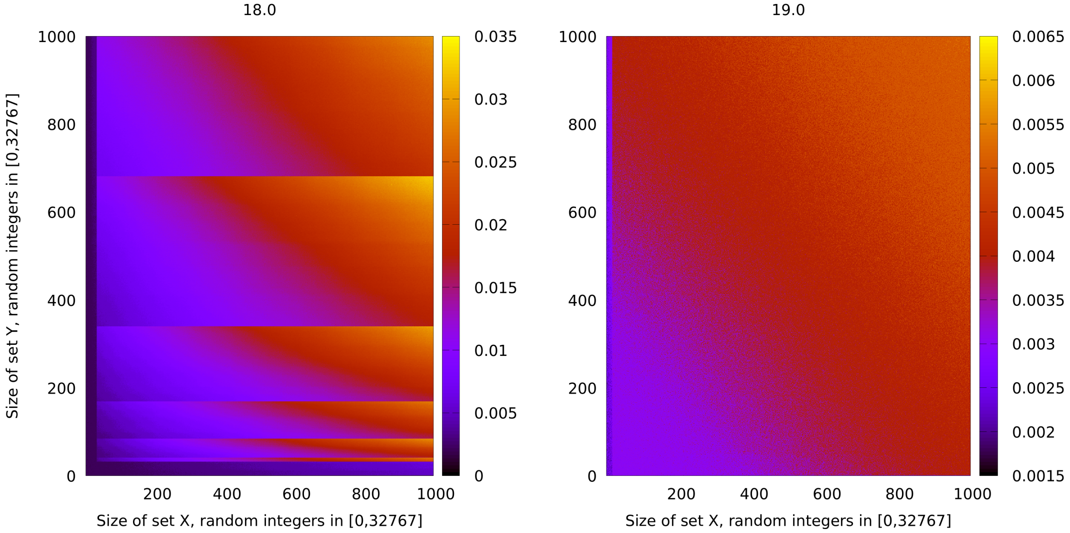
X∈Y — Total elapsed time / ms



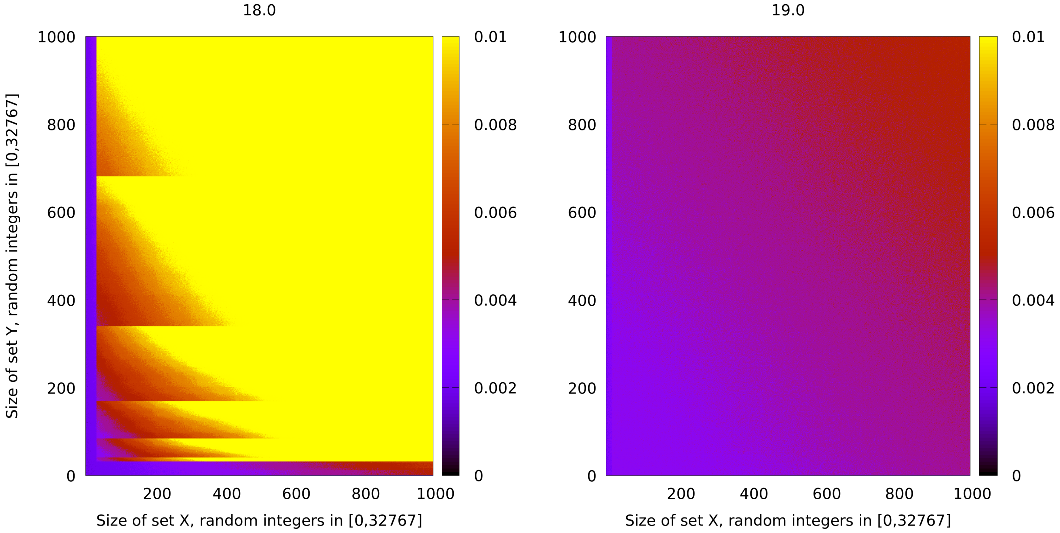
X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



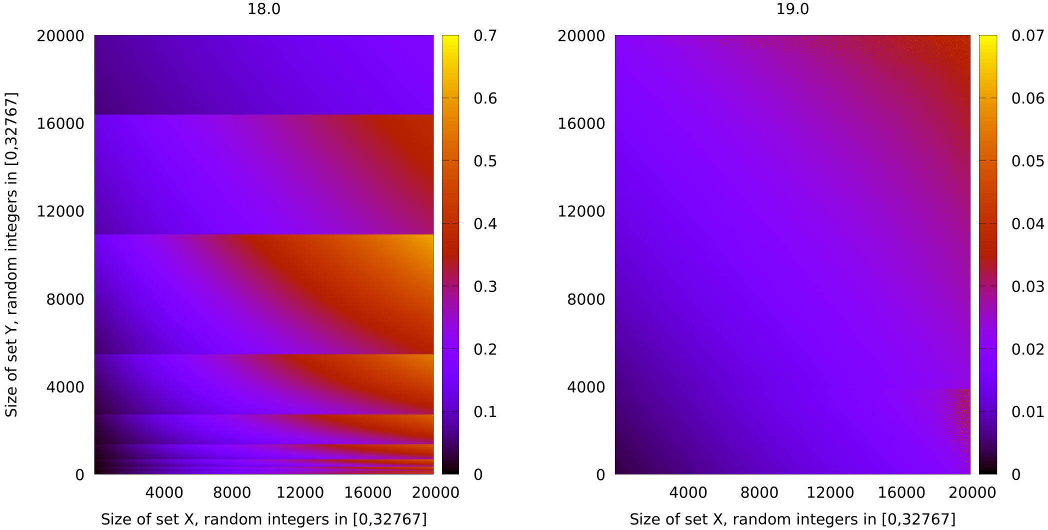
X∈Y — Total elapsed time / ms



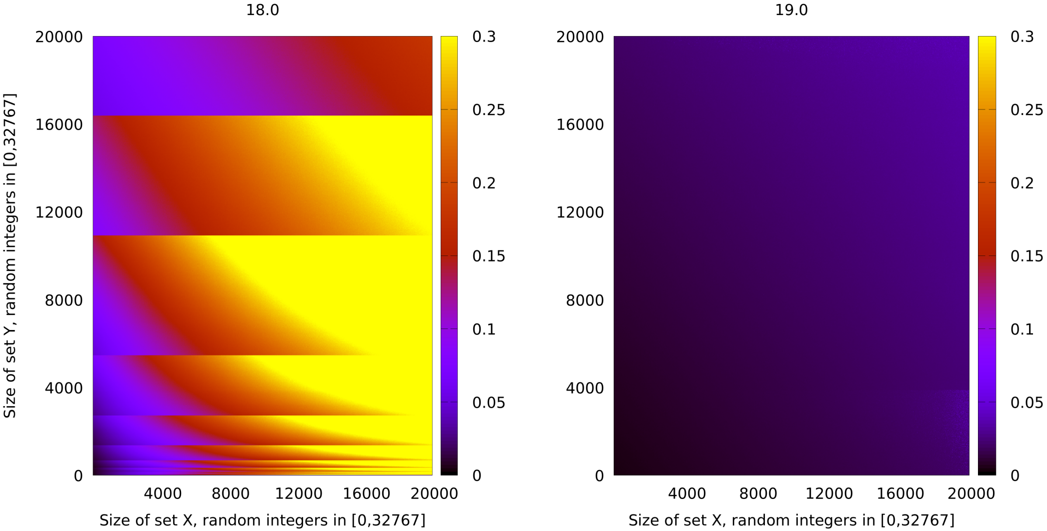
18.0

19.0

X∈Y — Total elapsed time / ms



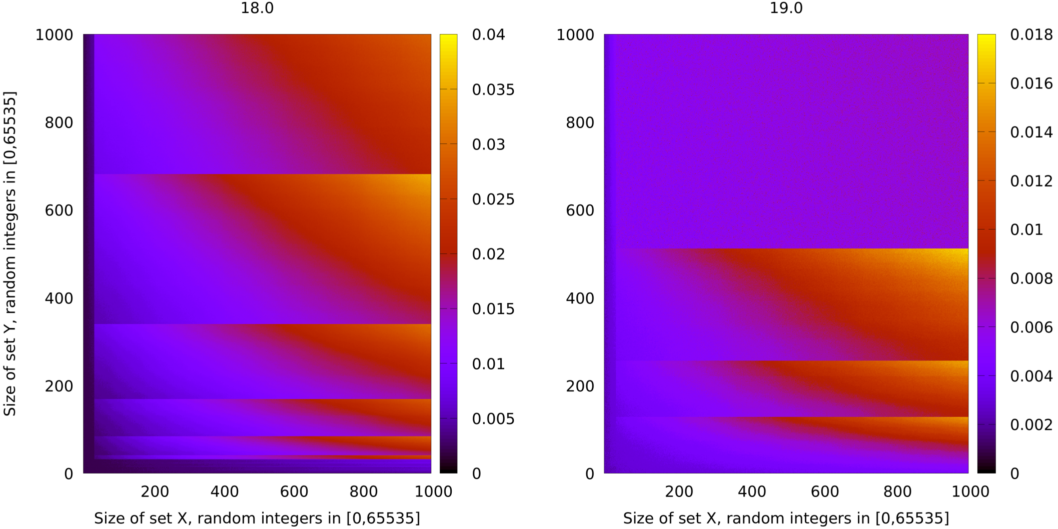
X∈Y — Total elapsed time / ms



18.0

19.0

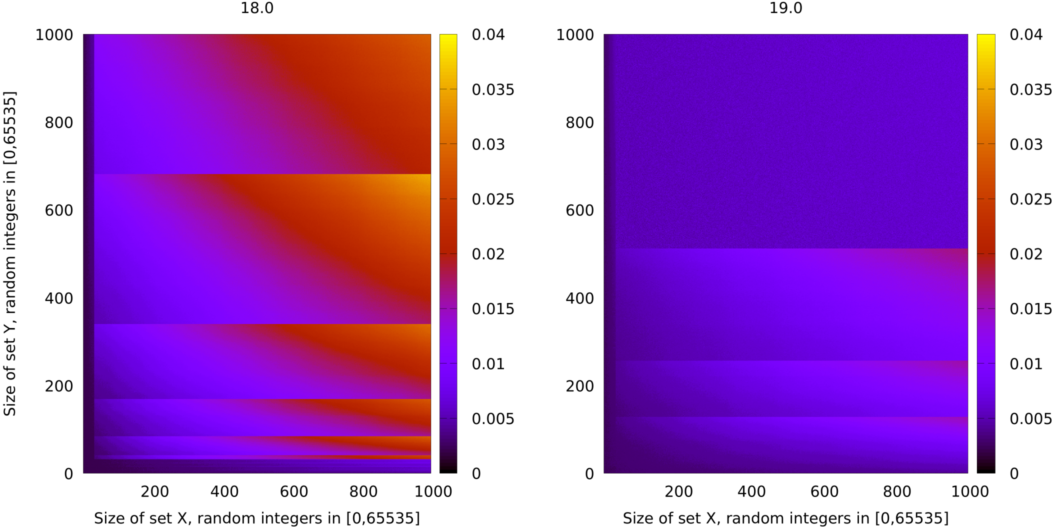
X∈Y — Total elapsed time / ms



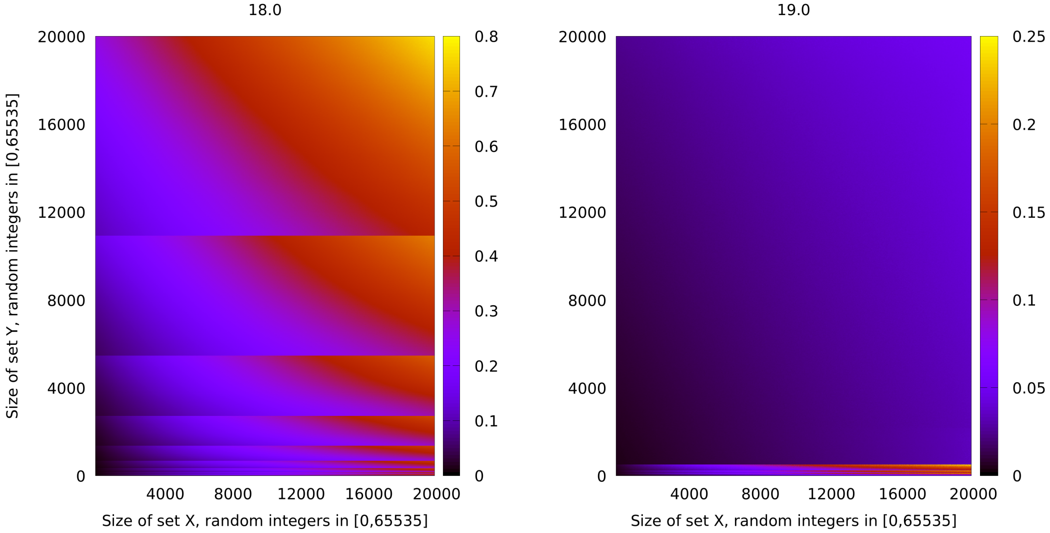
18.0

19.0

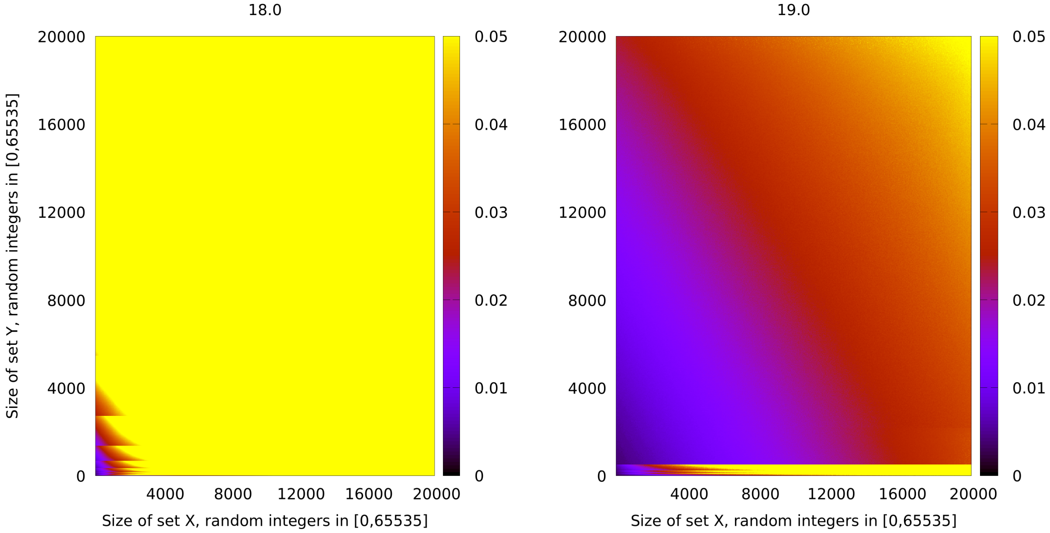
X∈Y — Total elapsed time / ms



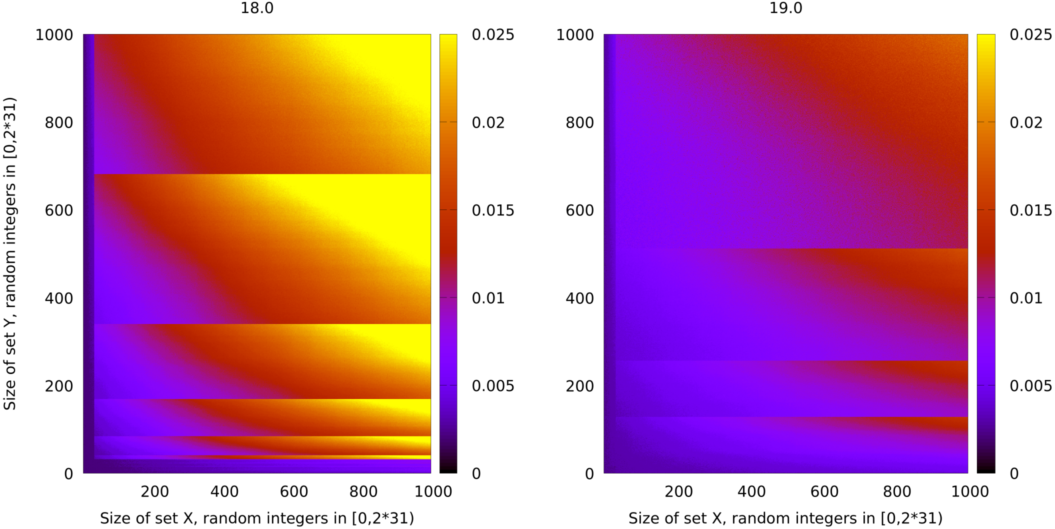
X∈Y — Total elapsed time / ms



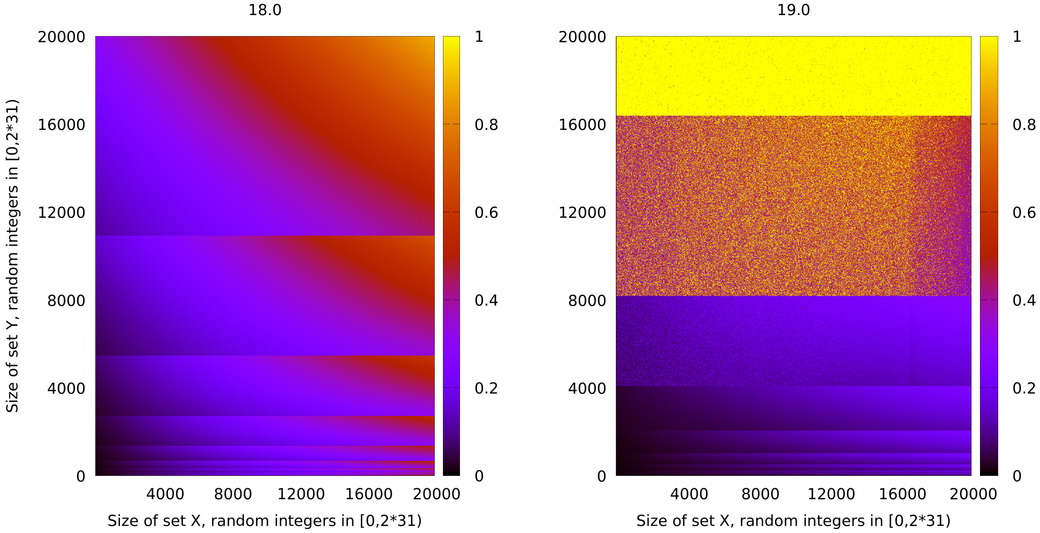
X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



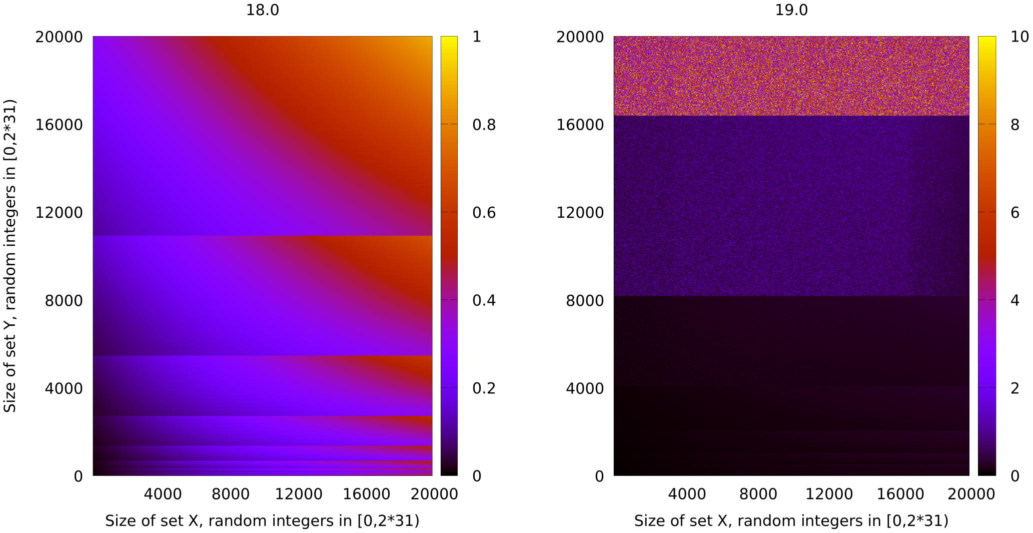
X∈Y — Total elapsed time / ms



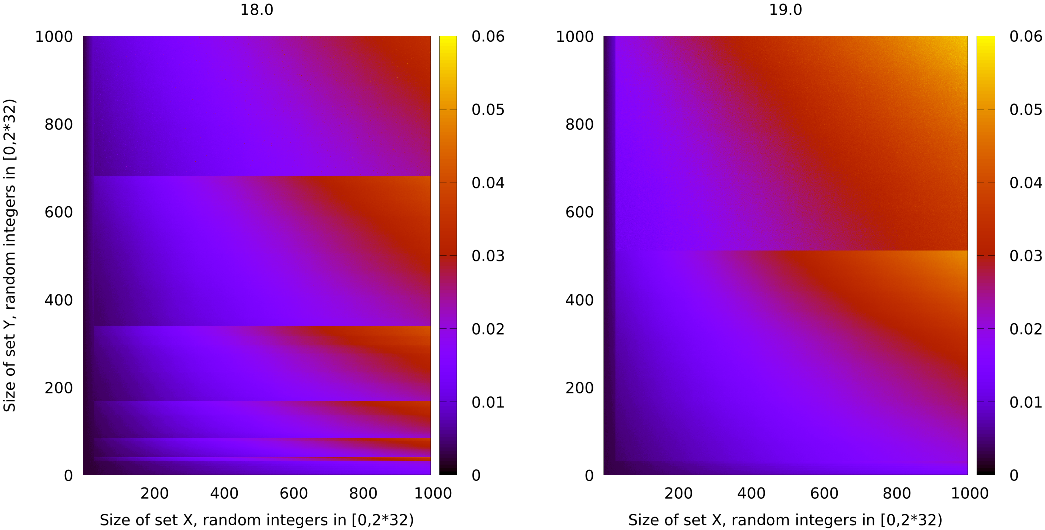
18.0

19.0

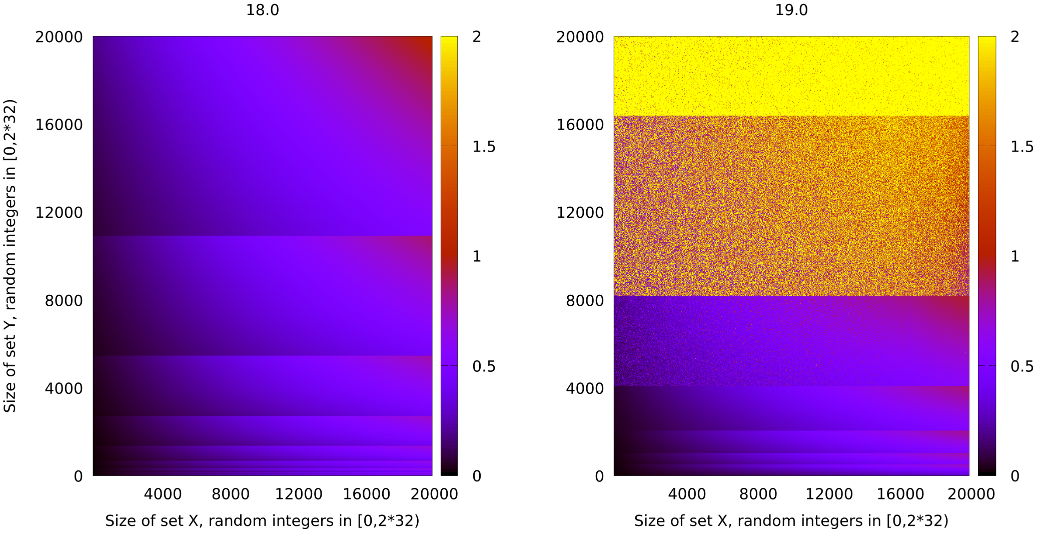
X∈Y — Total elapsed time / ms



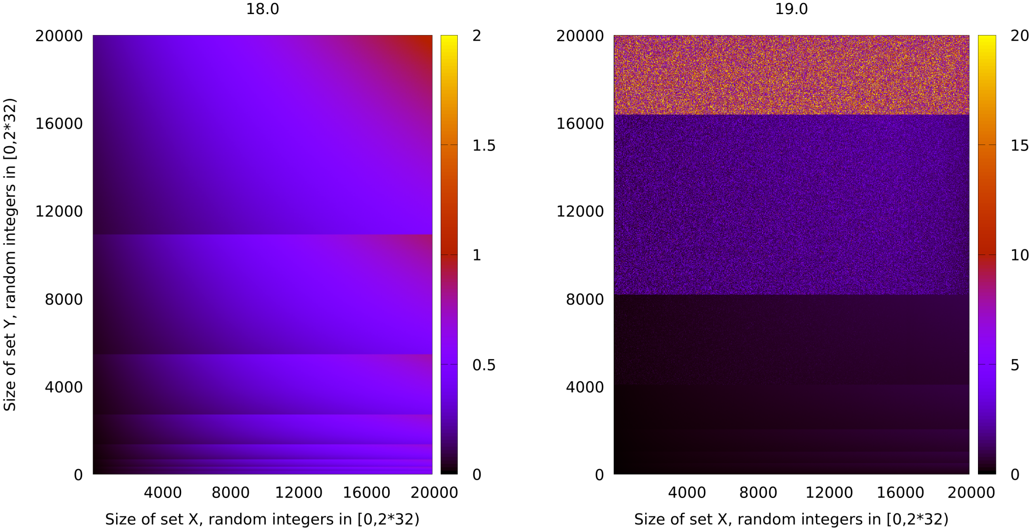
X∊Y — Total elapsed time / ms



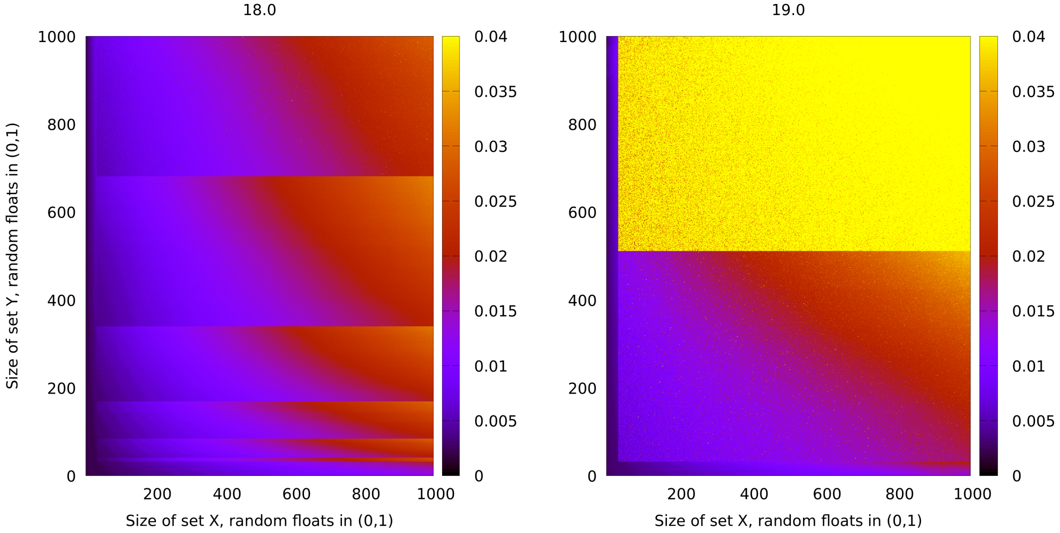
X∈Y — Total elapsed time / ms



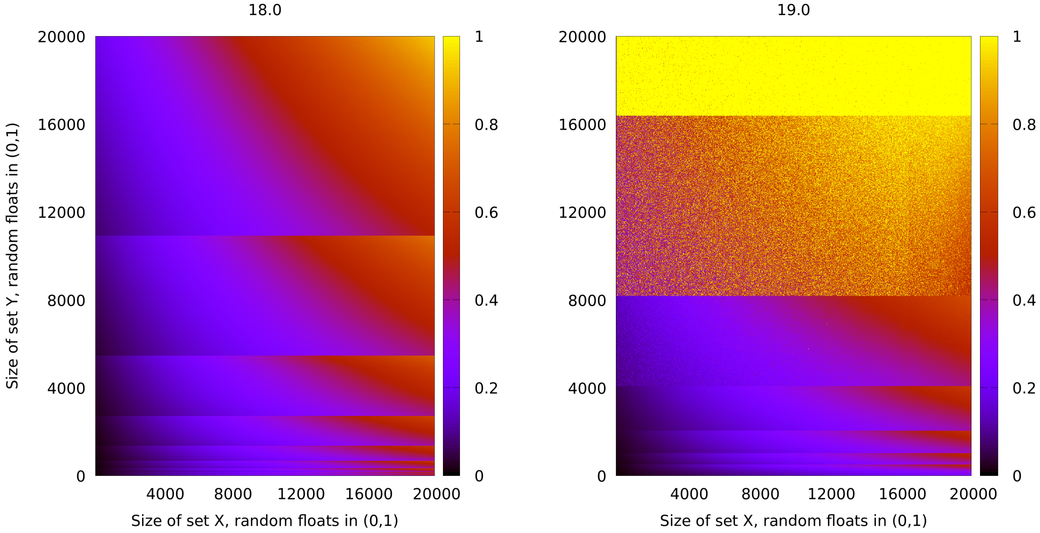
X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



Conclusions

Conclusions

- 19.0 often has better performance than 18.0...

Conclusions

- 19.0 often has better performance than 18.0...
- However there are cases where 18.0 performance is significantly better

Conclusions

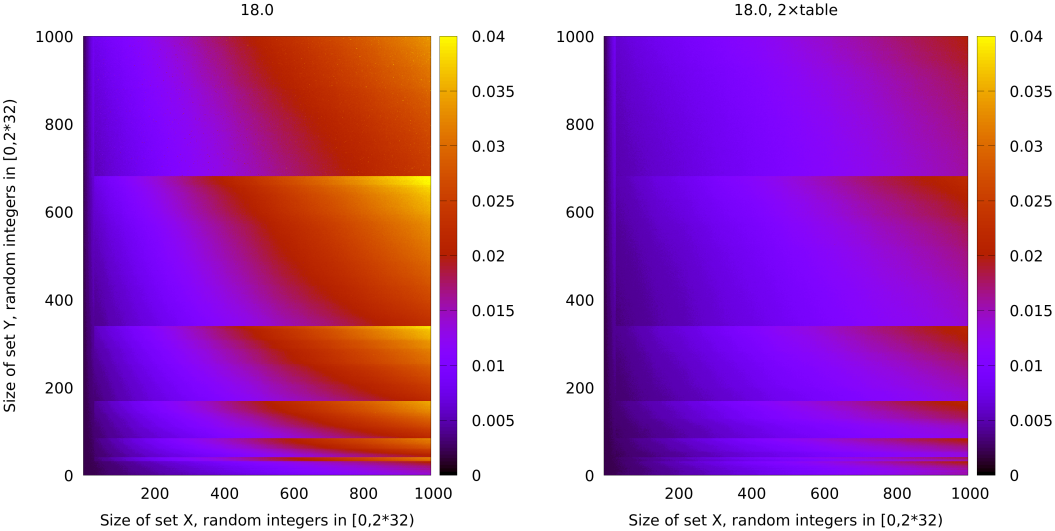
- 19.0 often has better performance than 18.0...
- However there are cases where 18.0 performance is significantly better
- Performance can be improved further in places

Future roadmap

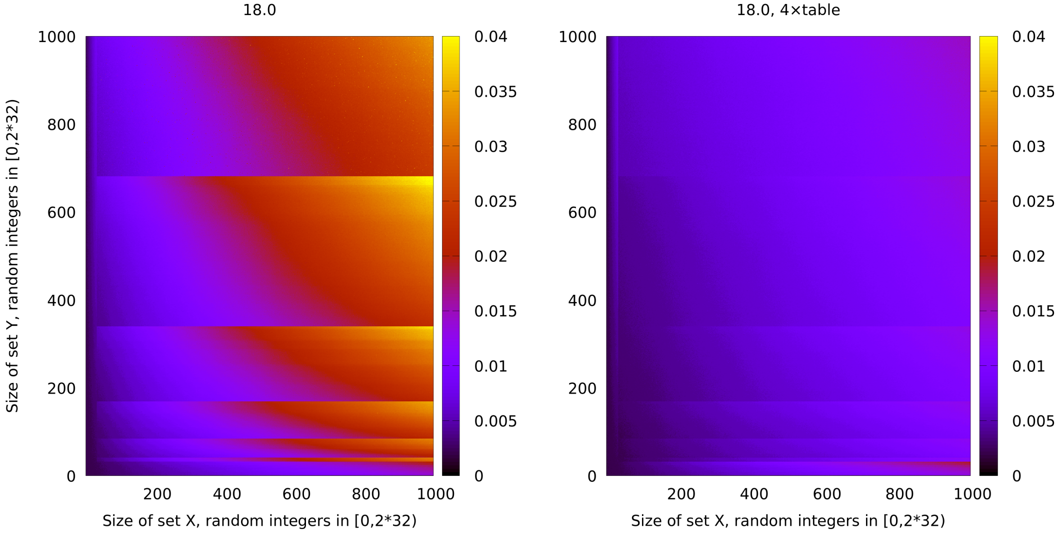
Future roadmap

- Hash table modifications, algorithm boundary changes

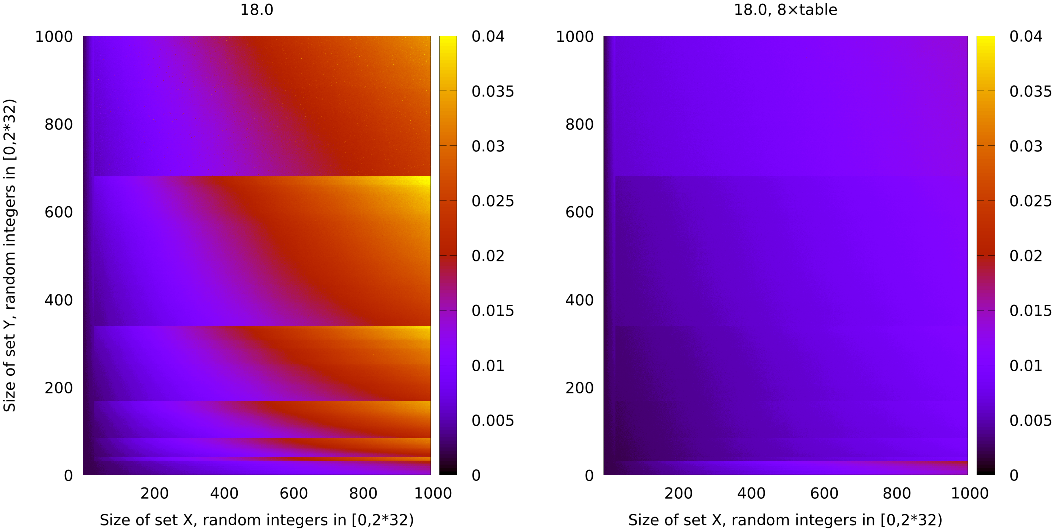
X∈Y — Total elapsed time / ms



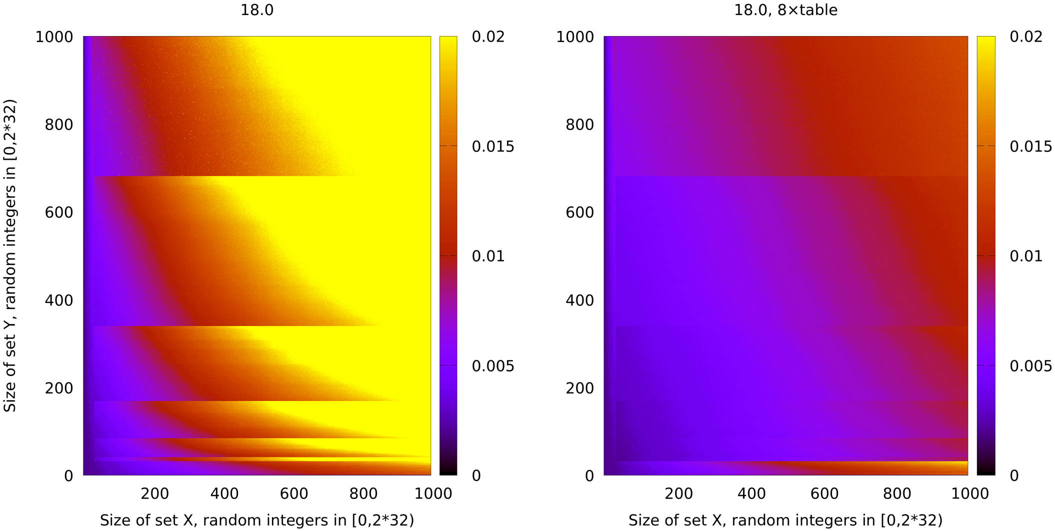
X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



X∈Y — Total elapsed time / ms



Future roadmap

- Hash table modifications, algorithm boundary changes

Future roadmap

- Hash table modifications, algorithm boundary changes
- Reintroduce successful 18.0 algorithms, code cleanup and reorganisation

Future roadmap

- Hash table modifications, algorithm boundary changes
- Reintroduce successful 18.0 algorithms, code cleanup and reorganisation
- Retained hash tables, hash algorithm modifications

Instrumentation

Instrumentation

- Increase internal hash table sizes (`8468⌶`): `8468⌶3`
- Increase internal lookup table sizes (`8469⌶`): `8469⌶16×1024*2`

Instrumentation

- Increase internal hash table sizes (`8468⌶`): `8468⌶3`
- Increase internal lookup table sizes (`8469⌶`): `8469⌶16×1024*2`
- Instrumented interpreter…