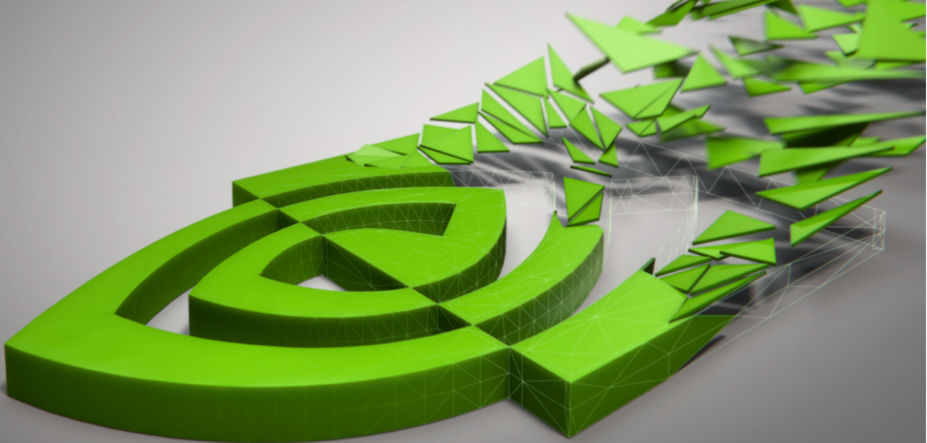
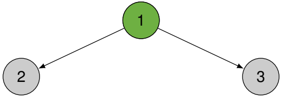
Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Nikolaus Binder and Alexander Keller, June, 2016



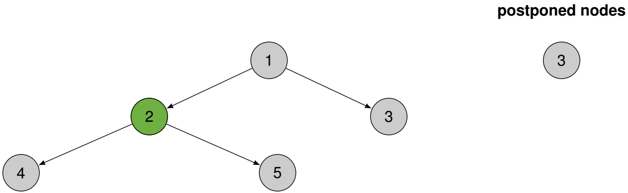
Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking



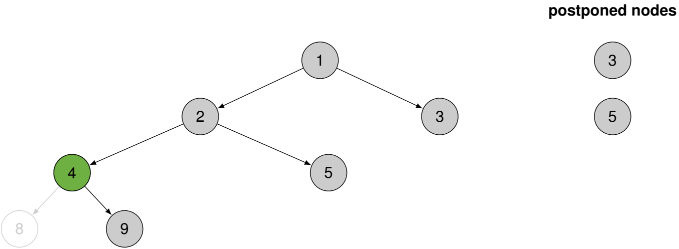
Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking



Efficient Hierarchy Traversal

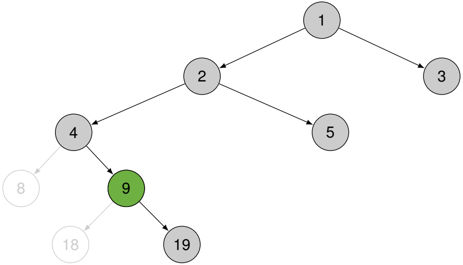
Pruning/postponing nodes and backtracking



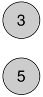
postponed nodes

Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking

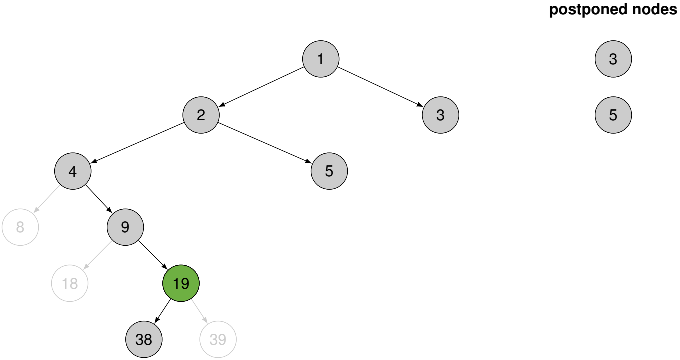


postponed nodes



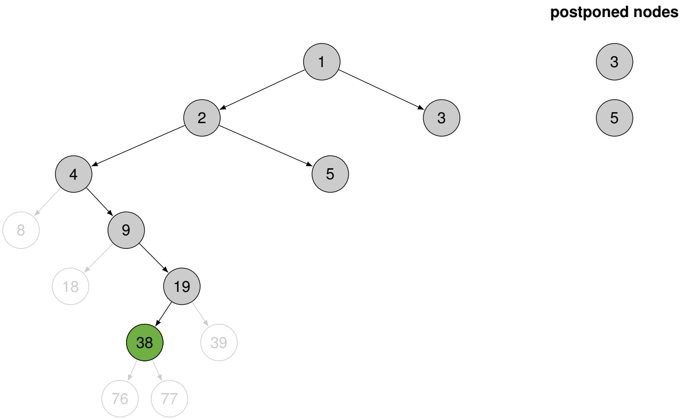
Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking



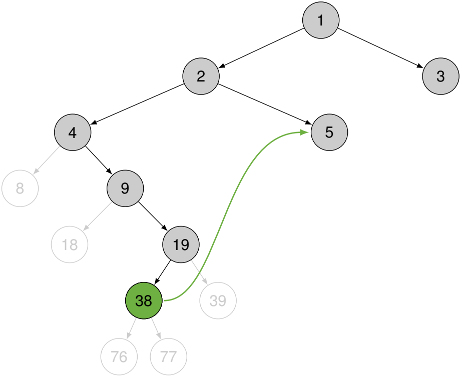
Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking

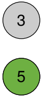


Efficient Hierarchy Traversal

Pruning/postponing nodes and backtracking

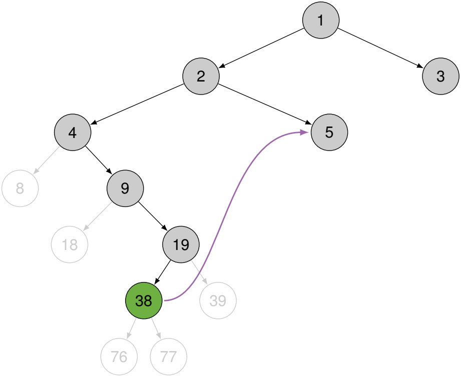


postponed nodes



Efficient Hierarchy Traversal

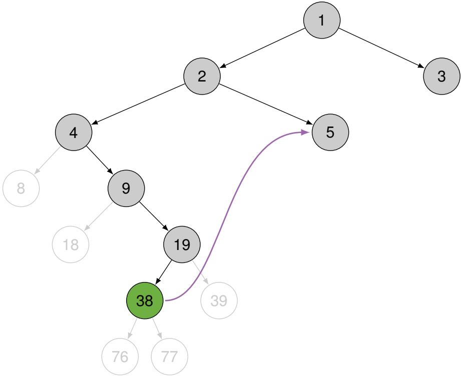
Comparing previous backtracking strategies



Stack
addr(3)
~~addr(5)~~

Efficient Hierarchy Traversal

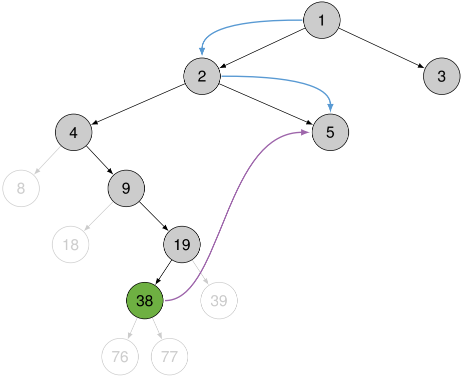
Comparing previous backtracking strategies



Stack	Bit Trail
addr(3)	
~~addr(5)~~	1
	1
	0
	0
	0

Efficient Hierarchy Traversal

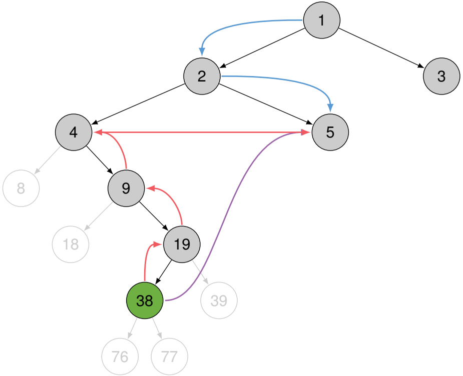
Comparing previous backtracking strategies



Stack	Bit Trail
addr(3)	
~~addr(5)~~	1
	~~1~~ 0
	~~0~~
	~~0~~
	~~0~~

Efficient Hierarchy Traversal

Comparing previous backtracking strategies



Stack	Bit Trail	Bit Trail
addr(3)		
~~addr(5)~~	1	1
	~~1~~ 0	~~1~~ 0
	~~0~~	~~0~~
	~~0~~	~~0~~
	~~0~~	~~0~~

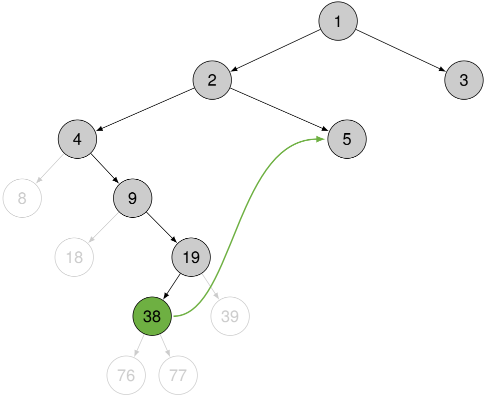
Efficient Hierarchy Traversal

Comparing previous backtracking strategies

	Stack	Stackless, Backtracking from root	Stackless, Backtracking with parents/siblings
state for book keeping (per ray)	$\mathcal{O}$(h(tree))	$\mathcal{O}$(1)	$\mathcal{O}$(1)
backtracking effort	$\mathcal{O}$(1)	$\mathcal{O}$(h(tree))	$\mathcal{O}$(h(tree))

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

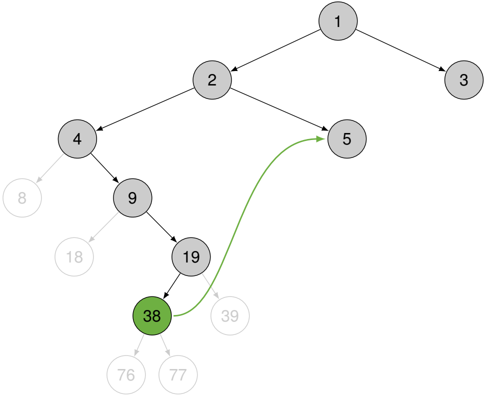
Building block 1: Using a bit trail, go to n^th uncle in constant time



Bit Trail	Cur Key
	1
1	0
1	0
0	1
0	1
0	0

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

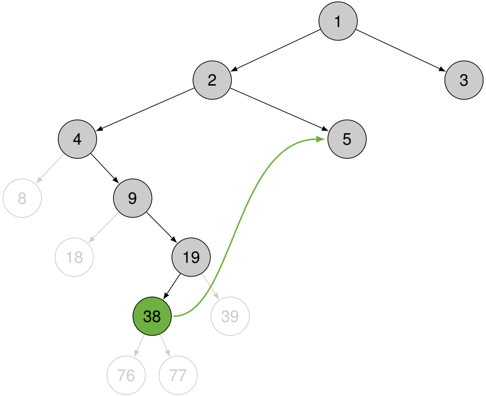
Building block 1: Using a bit trail, go to nth uncle in constant time



Bit Trail	Cur Key
	1
1	0
1	0
~~0~~	~~1~~
~~0~~	~~1~~
~~0~~	~~0~~

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

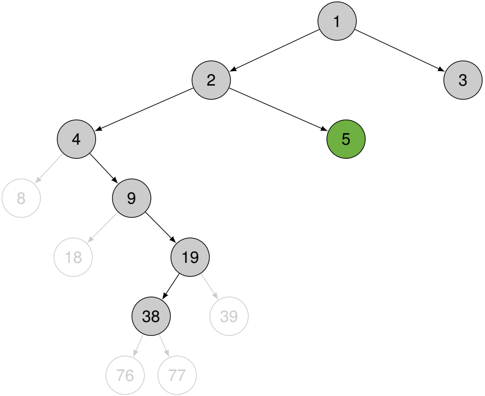
Building block 1: Using a bit trail, go to nth uncle in constant time



Bit Trail	Cur Key
	1
1	0
~~1~~ 0	~~0~~ 1

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 1: Using a bit trail, go to nth uncle in constant time



Bit Trail	Cur Key
	1
1	0
0	1

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 1: Using a bit trail, go to nth uncle in constant time

Perfect Hash Map h: node key $k \mapsto$ node address addr(k)

- properties
  - no collisions
  - no need to store keys
  - lookup in constant time

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

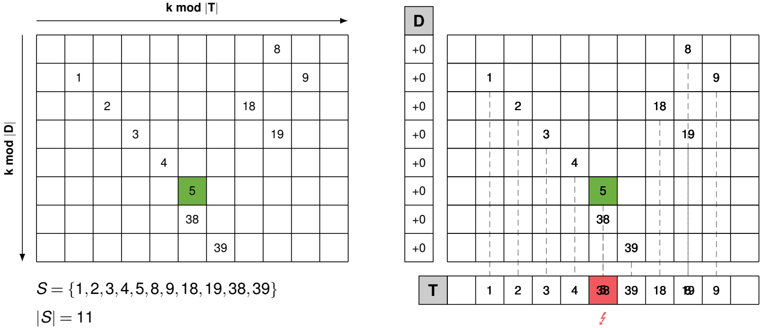
Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k\ mod\ |T| + D[k\ mod\ |D|])\ mod\ |T|$ [Tarjan, Yao 1979]

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]



								8		
	1								9	
		2					18			
			3					19		
				4						
					🟩 5					
					38					
						39				

D
+0
+0
+0
+0
+0
+0
+0
+0

T		1	2	3	4	🟥 5 38	39	18	8 19	9	

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]

k mod |T|

k mod |D|

											#	**D**
								8			1	+0
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+0
						39					1	+0

T											

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

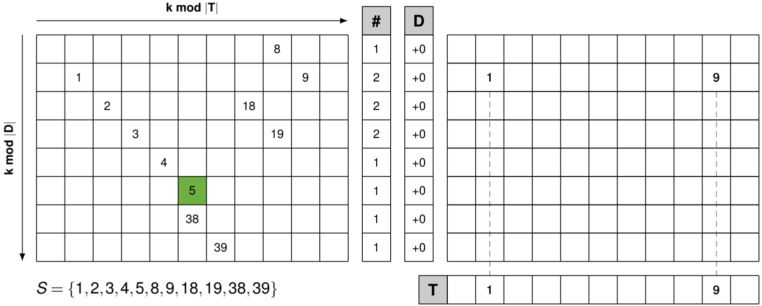
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



k mod |T|

k mod |D|

											#	D
								8			1	+0
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+0
						39					1	+0

	1								9	

T		1								9	

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

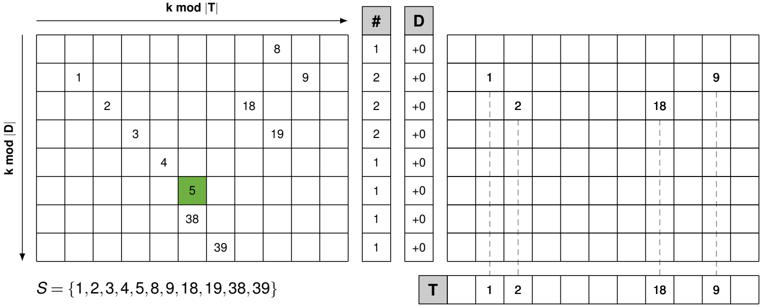
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+0
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+0
						39					1	+0

	1								9	
		2					18			

T											
		1	2					18		9	

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

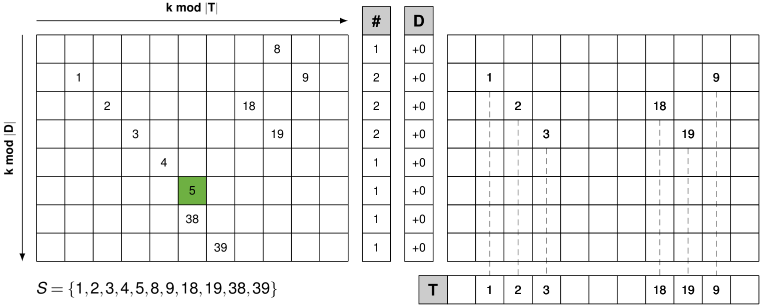
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+0
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+0
						39					1	+0

	1								9	
		2					18			
			3					19		

T											
		1	2	3				18	19	9	

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

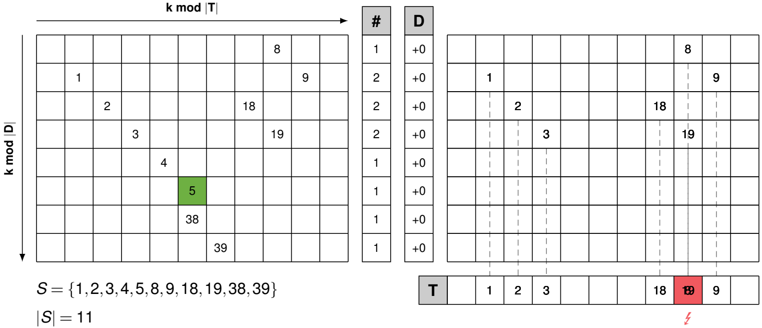
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \ mod \ |T| + D[k \ mod \ |D|]) \ mod \ |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+0
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					🟩 5						1	+0
					38						1	+0
						39					1	+0

								8		
	1								9	
		2					18			
			3					19		

T		1	2	3				18	🟥 8 19	9	
									↯		

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

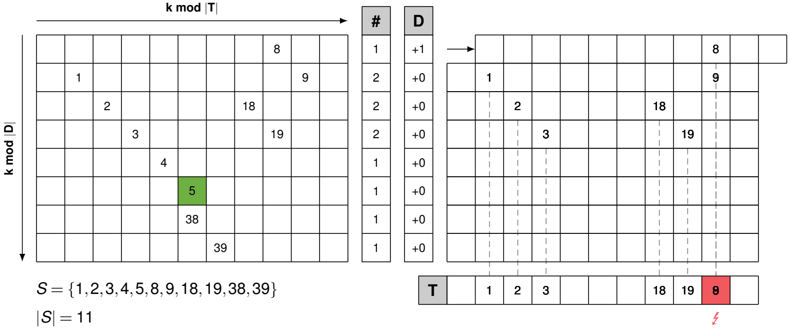
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \ mod \ |T| + D[k \ mod \ |D|]) \ mod \ |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+1
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					🟩 5						1	+0
					38						1	+0
						39					1	+0

										8	
	1								9		
		2					18				
			3					19			

T		1	2	3				18	19	🟥 8 9	

$\lightning$

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

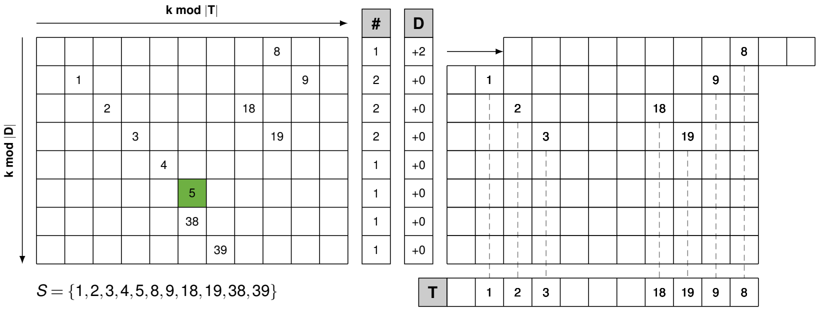
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k\ mod\ |T| + D[k\ mod\ |D|])\ mod\ |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

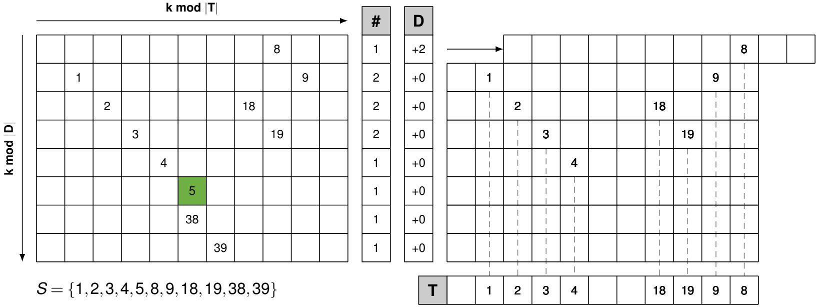
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \bmod |T| + D[k \bmod |D|]) \bmod |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					🟩 5						1	+0
					38						1	+0
						39					1	+0

T		1	2	3	4			18	19	9	8

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

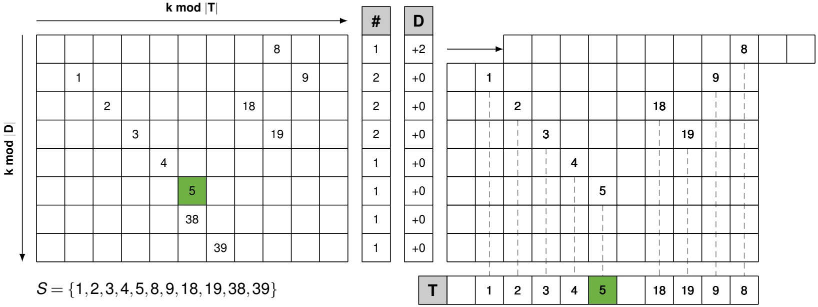
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+0
						39					1	+0

T		1	2	3	4	5		18	19	9	8

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

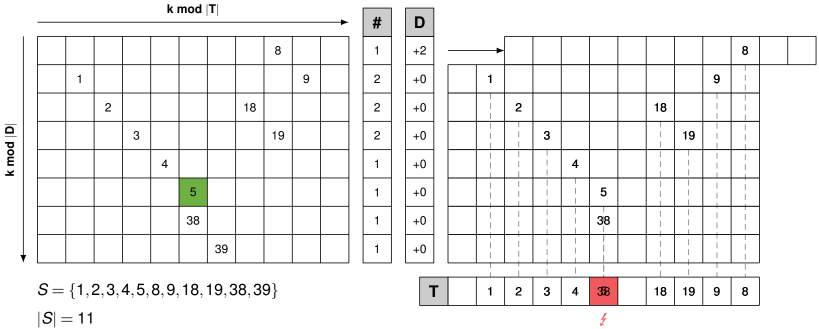
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

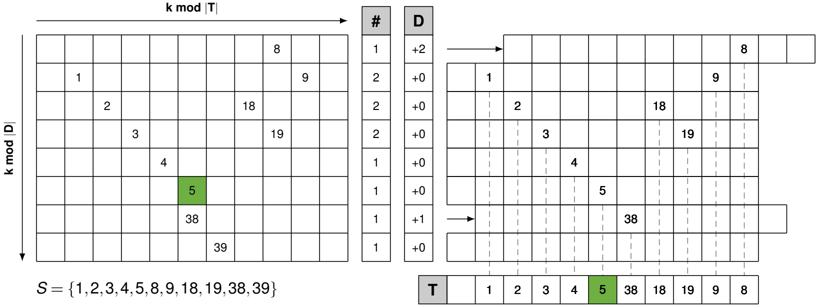
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \bmod |T| + D[k \bmod |D|]) \bmod |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+1
						39					1	+0

										8		
	1								9			
		2					18					
			3					19				
				4								
					5							
						38						

T		1	2	3	4	5	38	18	19	9	8

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

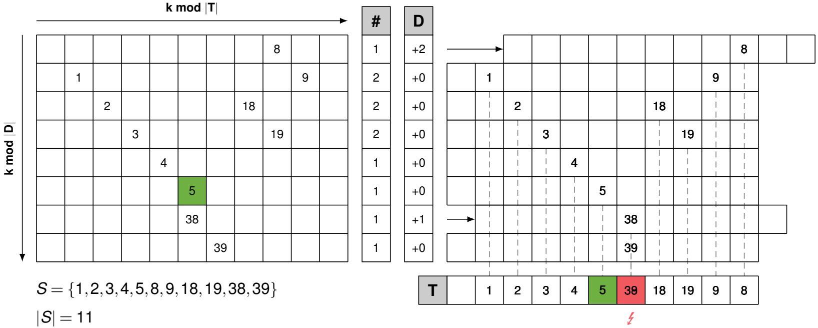
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					🟩 5						1	+0
					38						1	+1
						39					1	+0

T		1	2	3	4	🟩 5	🟥 39	18	19	9	8
							↯				

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

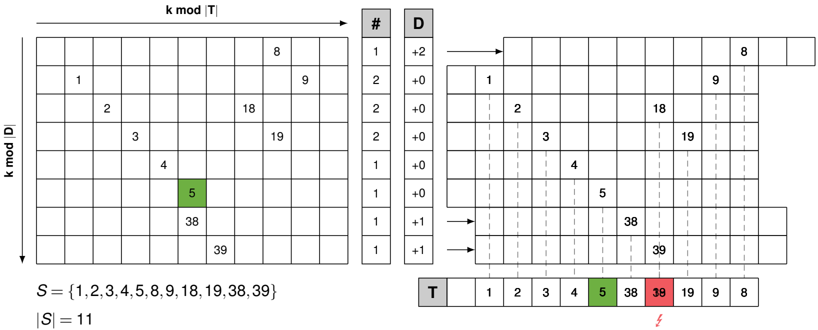
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \bmod |T| + D[k \bmod |D|]) \bmod |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

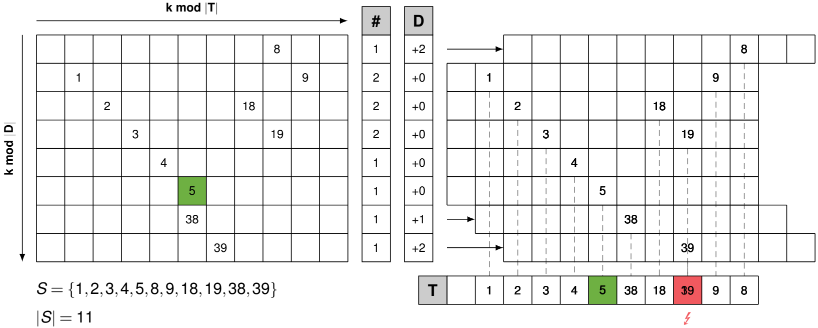
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \ mod \ |T| + D[k \ mod \ |D|]) \ mod \ |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+1
						39					1	+2

T		1	2	3	4	5	38	18	39	9	8

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

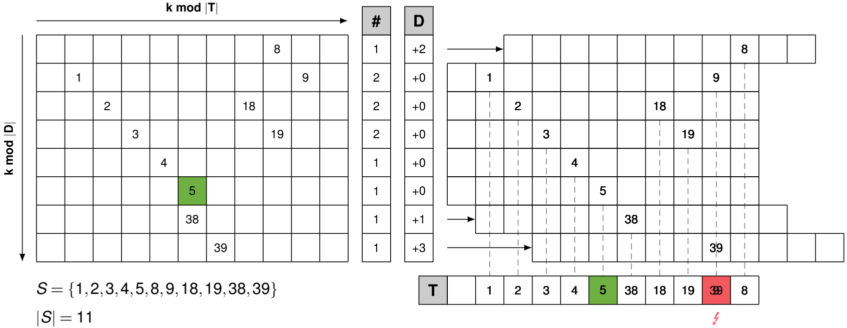
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k\ mod\ |T| + D[k\ mod\ |D|])\ mod\ |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+1
						39					1	+3

T		1	2	3	4	5	38	18	19	39 9	8

$S = \{1,2,3,4,5,8,9,18,19,38,39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

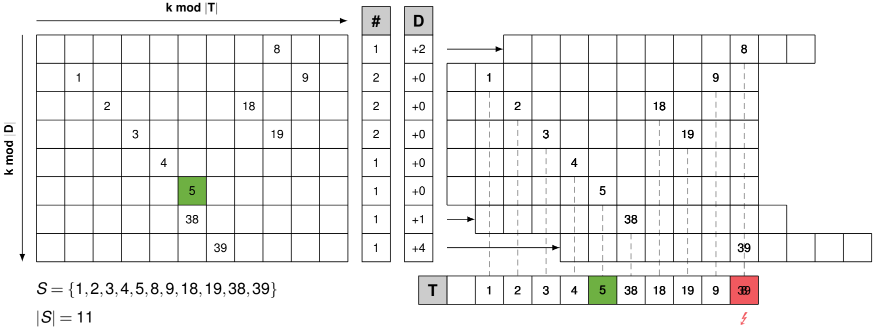
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \; mod \; |T| + D[k \; mod \; |D|]) \; mod \; |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+1
						39					1	+4

T		1	2	3	4	5	38	18	19	9	38 39

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

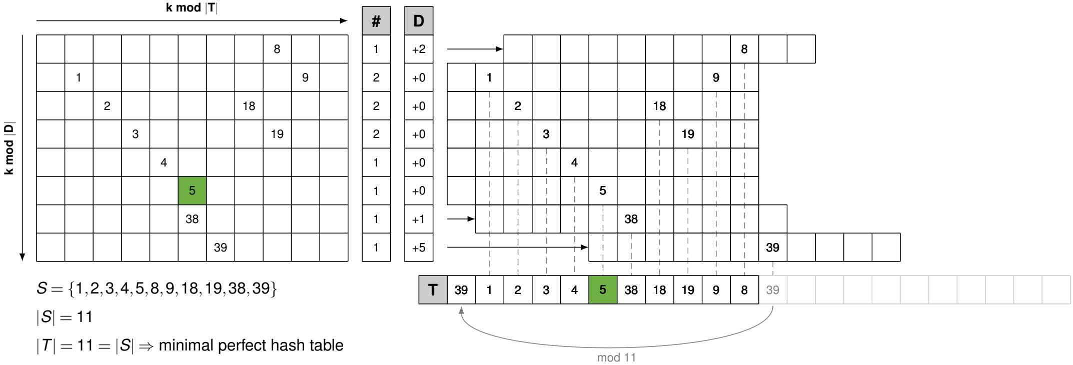
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 2: Two level hashing using an additional displacement table D

$k \mapsto (k \bmod |T| + D[k \bmod |D|]) \bmod |T|$ [Tarjan, Yao 1979]

Greedy resolution in decreasing number of dependencies [Fox, Heath, Chen, and Daoud 1992]



											#	D
								8			1	+2
	1								9		2	+0
		2					18				2	+0
			3					19			2	+0
				4							1	+0
					5						1	+0
					38						1	+1
						39					1	+5

T	39	1	2	3	4	5	38	18	19	9	8	39										

$S = \{1, 2, 3, 4, 5, 8, 9, 18, 19, 38, 39\}$

$|S| = 11$

$|T| = 11 = |S| \Rightarrow$ minimal perfect hash table

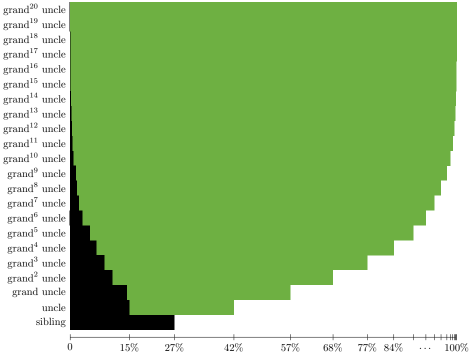
$|D| = 8$

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 3: Reducing the number of hash lookups

- backtracking statistics
  - to sibling: 27%
  - to uncle: 15%
  - to grand uncle: 15%

around 57% alltogether



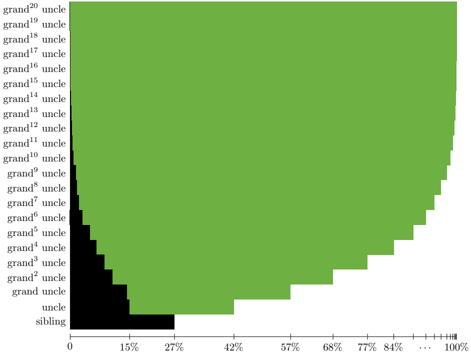
Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 3: Reducing the number of hash lookups

- backtracking statistics
  - to sibling: 27%
  - to uncle: 15%
  - to grand uncle: 15%

  } around 57% alltogether

- store references to uncle and grand uncle in node
  - in unused padding space
  - data loaded anyway



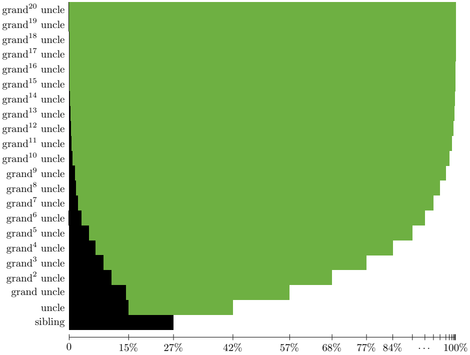
Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 3: Reducing the number of hash lookups

- backtracking statistics
  - to sibling: 27%
  - to uncle: 15%
  - to grand uncle: 15%

  } around 57% alltogether

- store references to uncle and grand uncle in node
  - in unused padding space
  - data loaded anyway
- store most recently postponed node in a register
  - always used for transitions to siblings
  - similar to a short stack, but more powerful



Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

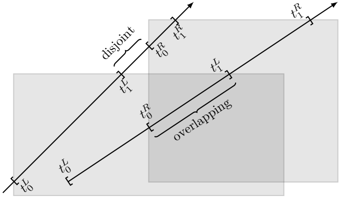
Building block 4: Avoid pointless backtracking

- subtrees behind intersection may not always be culled
  - due to overlapping bounding boxes

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 4: Avoid pointless backtracking

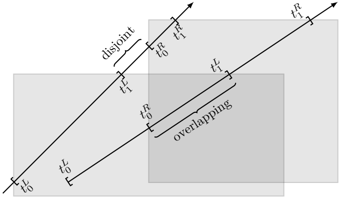
- subtrees behind intersection may not always be culled
  - due to overlapping bounding boxes
- discard levels with disjoint *t*-intervals



Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 4: Avoid pointless backtracking

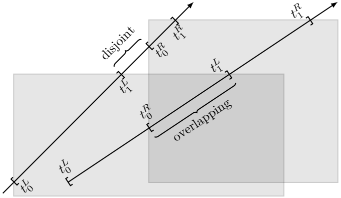
- subtrees behind intersection may not always be culled
  - due to overlapping bounding boxes
- discard levels with disjoint t-intervals
  - cheap
    - no t_0 values stored
    - mask with one bit per level
    - bit set to one if overlapping, zero if disjoint
    - bitwise `and` with bit trail after intersection has been found



Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 4: Avoid pointless backtracking

- subtrees behind intersection may not always be culled
  - due to overlapping bounding boxes
- discard levels with disjoint t-intervals
  - cheap
    - no t_0 values stored
    - mask with one bit per level
    - bit set to one if overlapping, zero if disjoint
    - bitwise `and` with bit trail after intersection has been found
  - compromise
    - cannot account for intersections outside overlap



Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Building block 5: Resuming traversal in last node node instead of starting at the root

- **pause:** state (key and bit trail) must be stored
- **resume:** start in last node, set bit trail to
  - previous bit trail if same ray origin and direction
    - transparent/translucent object, cut outs
  - 1 for all levels above current level if ray origin or direction has changed
    - tracing paths
    - refraction

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Summary

- optimized stackless traversal
  - backtracking in constant time by perfect hashing
  - reduced number of hash lookups
    - store references to uncles and grand uncles in nodes
    - store most recently postponed node in a register

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Summary

- optimized stackless traversal
  - backtracking in constant time by perfect hashing
  - reduced number of hash lookups
    - store references to uncles and grand uncles in nodes
    - store most recently postponed node in a register
- additional building blocks currently not used in software (e.g. due to register pressure)
  - discard unreachable postponed nodes
  - pause and resume traversal in current node

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Summary

- optimized stackless traversal
  - backtracking in constant time by perfect hashing
  - reduced number of hash lookups
    - store references to uncles and grand uncles in nodes
    - store most recently postponed node in a register
- additional building blocks currently not used in software (e.g. due to register pressure)
  - discard unreachable postponed nodes
  - pause and resume traversal in current node
- exhaustive tests
  - many different and freely available scenes
  - various practical camera positions
  - different ray types

Efficient Stackless Hierarchy Traversal with Backtracking in Constant Time

Results: Performance in M rays/s, NVIDIA Titan X, for Primary/Shadow/Diffuse Rays

	Stack [Aila 2009]			Stackless [Áfra 2014]			ours		
	Primary	Shadow	Diffuse	P	S	D	P	S	D
Armadillo	837	236	214	-13%	-10%	-11%	+17%	+32%	+35%
Conference	786	399	253	-16%	-2%	-13%	+4%	+25%	+20%
Dragon	743	212	194	-16%	-13%	-15%	+17%	+32%	+31%
Emily	676	254	234	-20%	-12%	-14%	+9%	+26%	+25%
Buddha	1237	210	185	-12%	-11%	-12%	+15%	+34%	+32%
Hairball	190	77	65	-23%	-6%	-12%	+1%	+25%	+22%
Enchanted Forest	237	81	64	-14%	-5%	-12%	+5%	+22%	+19%
San-Miguel	246	149	81	-20%	-7%	-20%	+4%	+23%	+10%
Average				**-17%**	**-12%**	**-19%**	**+8%**	**+20%**	**+17%**

We are hiring.

akeller@nvidia.com