

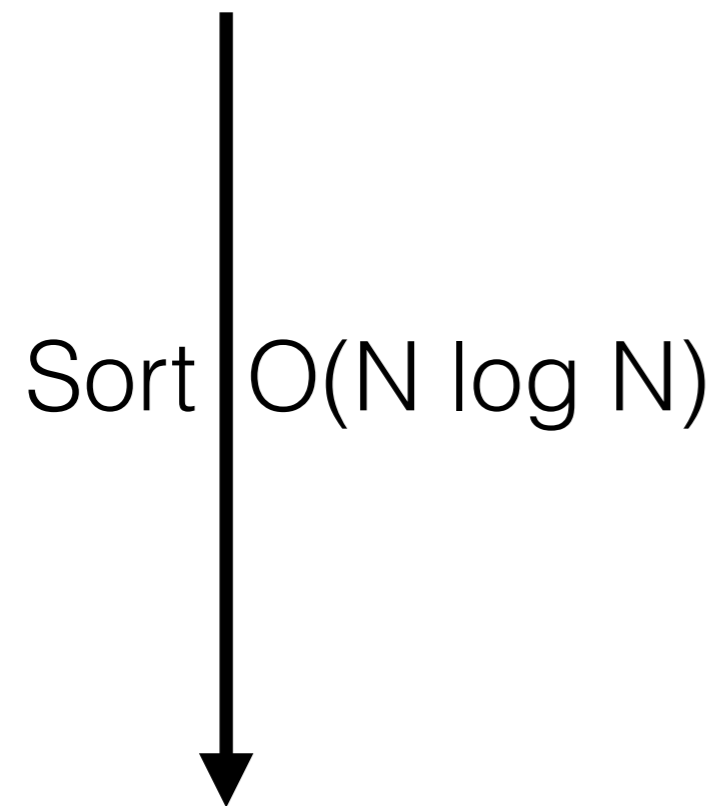
Towards Pervasive Incremental Computing

What is Incremental Computation?

A program is incremental if repeating it with a changed input is faster than from-scratch recomputation

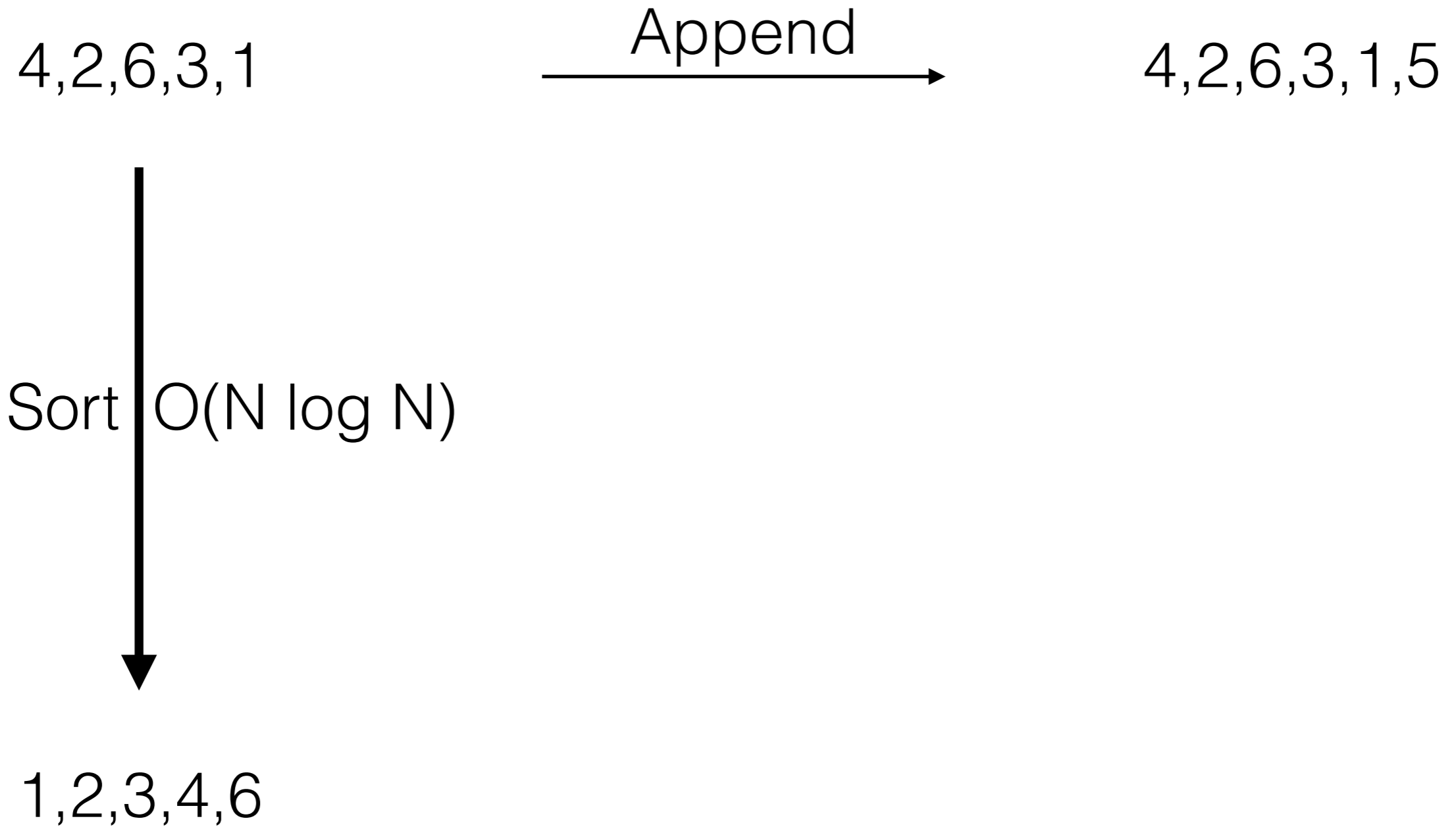
What is Incremental Computation?

4,2,6,3,1



1,2,3,4,6

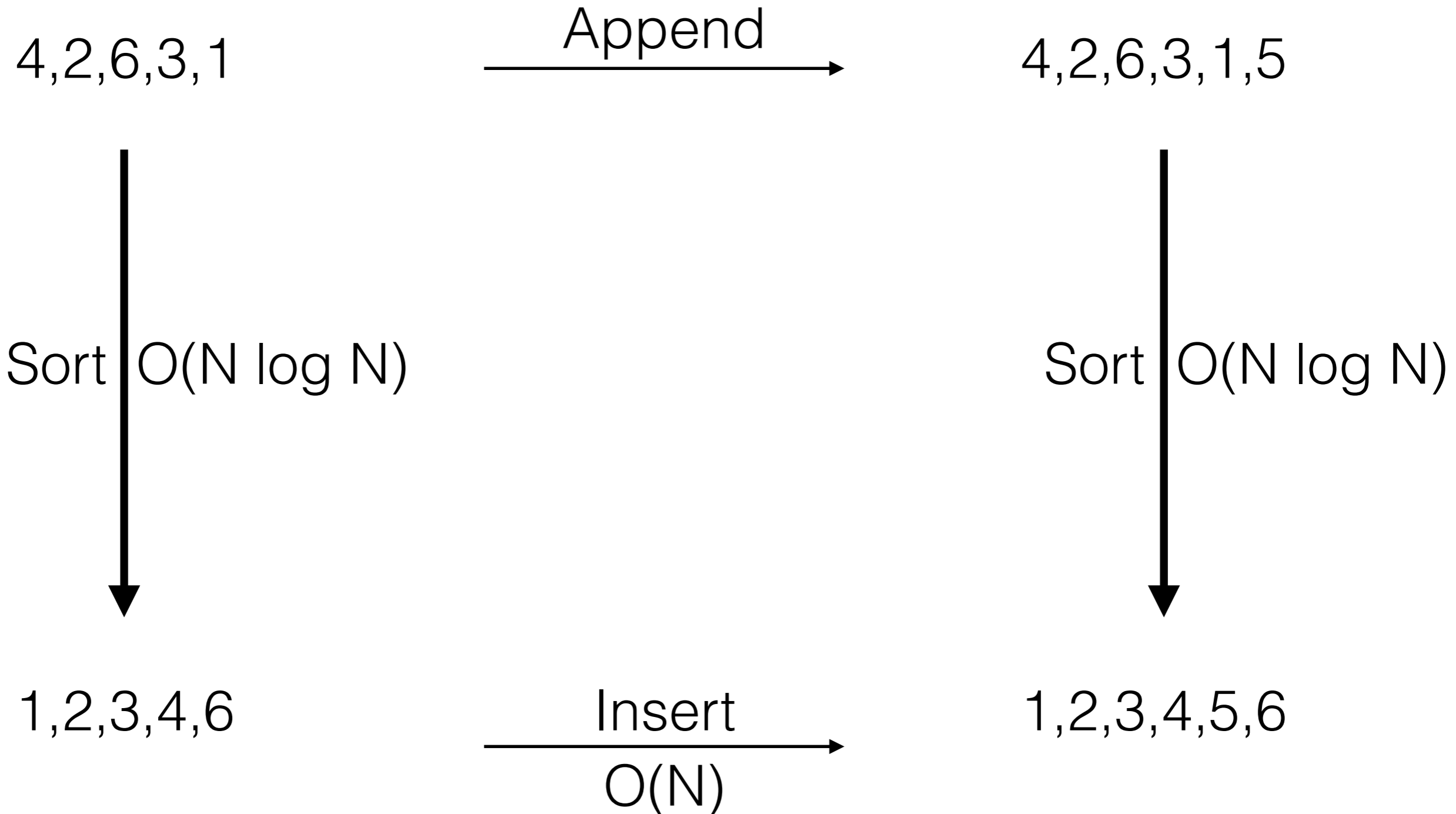
What is Incremental Computation?



What is Incremental Computation?



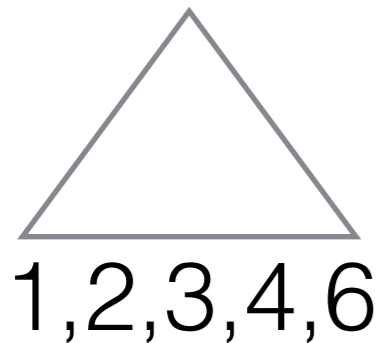

What is Incremental Computation?



What is Incremental Computation?

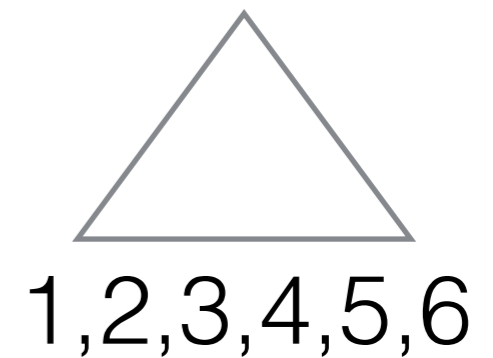
4,2,6,3,1

Insert into Heap
 $O(N \log N)$



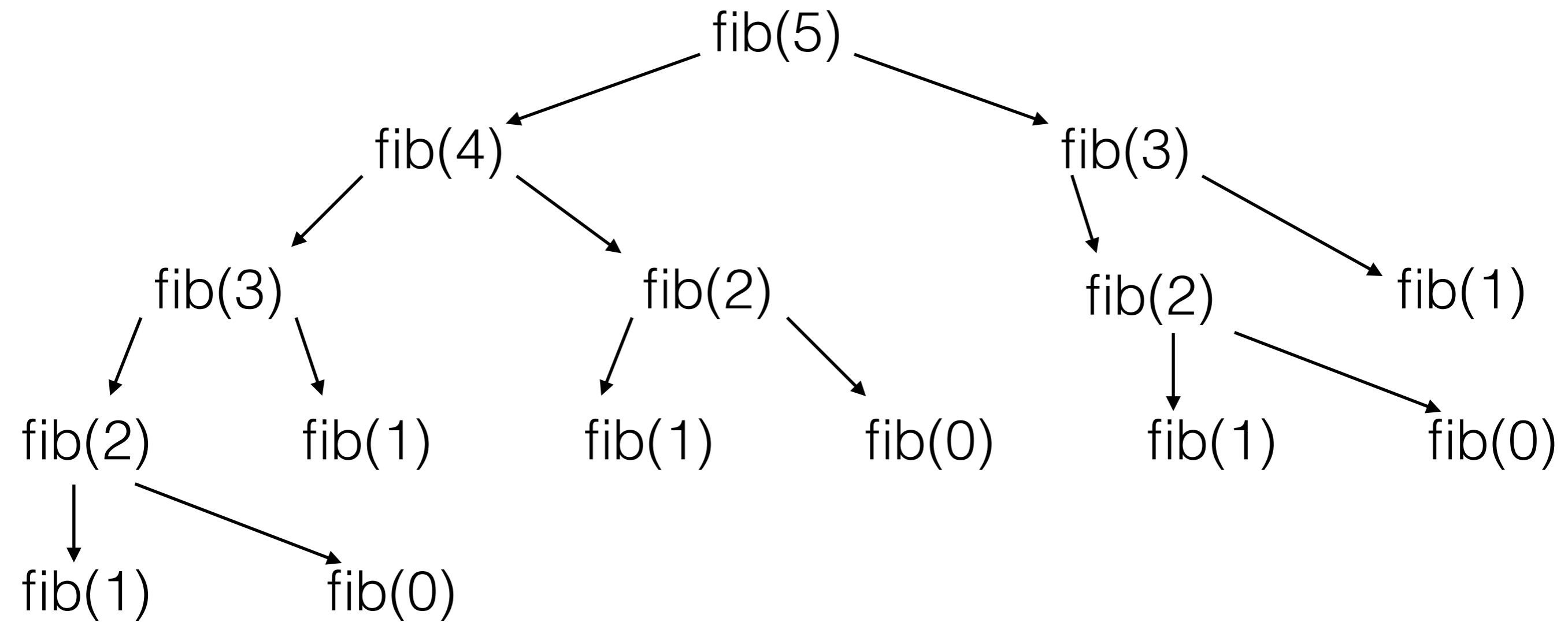
Using a specialized data structure allows incremental sorting

Insert
 $O(\log N)$



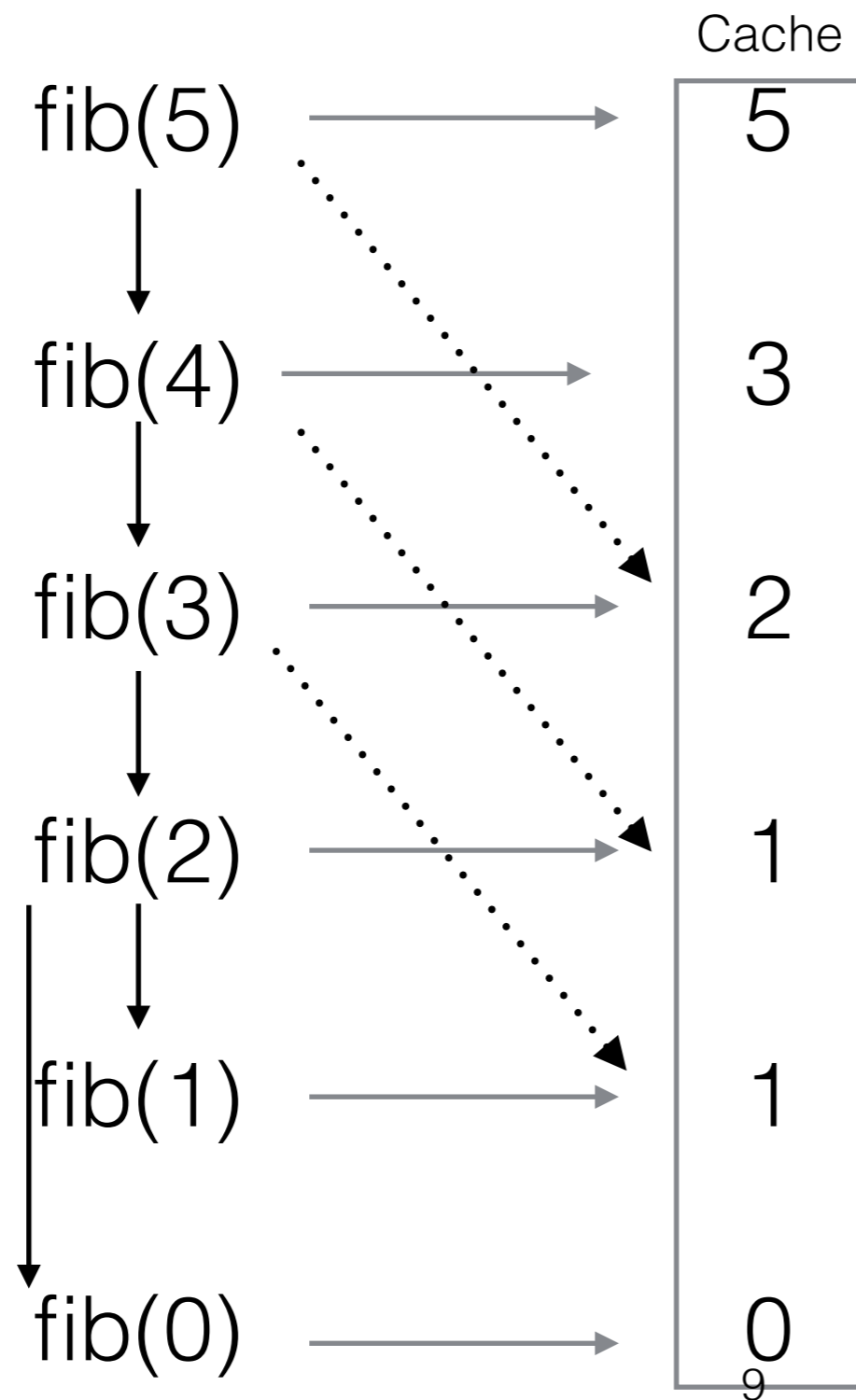
What is Incremental Computation?

Computing a Fibonacci number



What is Incremental Computation?

Computing a Fibonacci number



Employing
caches allows
incremental
functions

Expanding Incremental Computation

Our goal is to provide access to incremental computation for all programs

Expanding Incremental Computation

Our goal is to provide access to incremental computation for all programs

Do these scale?

Are they accurate?

caching

reactive programming

decorators

libraries

invertible functions

incremental data structures

function derivatives

languages

Expanding Incremental Computation

Our goal is to provide access to incremental computation for all programs

Do these scale?

Are they accurate?

caching

decorators

invertible functions

function derivatives

reactive programming

libraries

incremental data structures

languages

Adapton

Build and maintain a partial-order call graph of cached results

On-demand re-evaluation

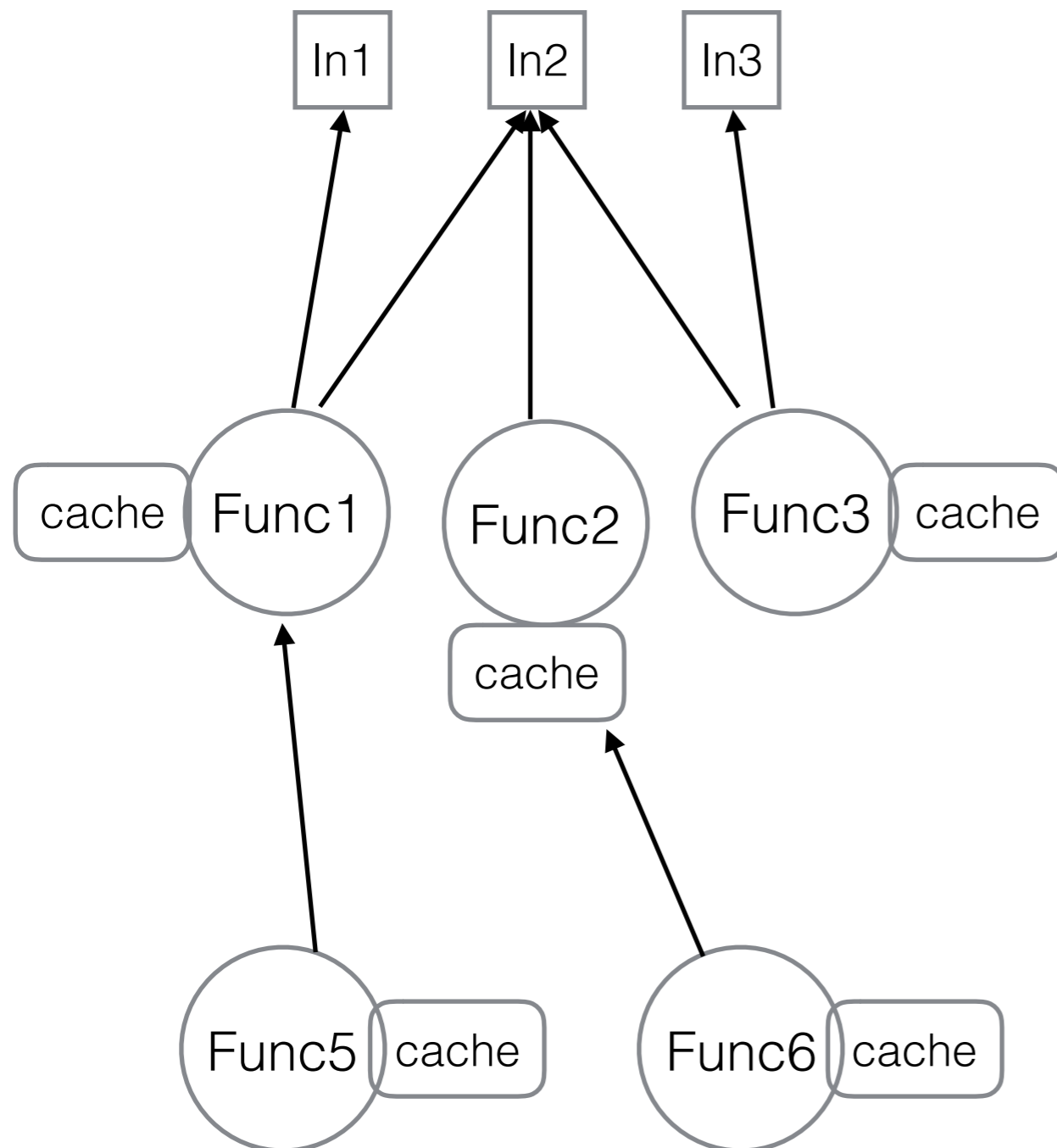
Minimum update paths

From-scratch consistent

Dynamic call-graph

Adapton

Build and maintain a partial-order call graph of cached results



On-demand re-evaluation

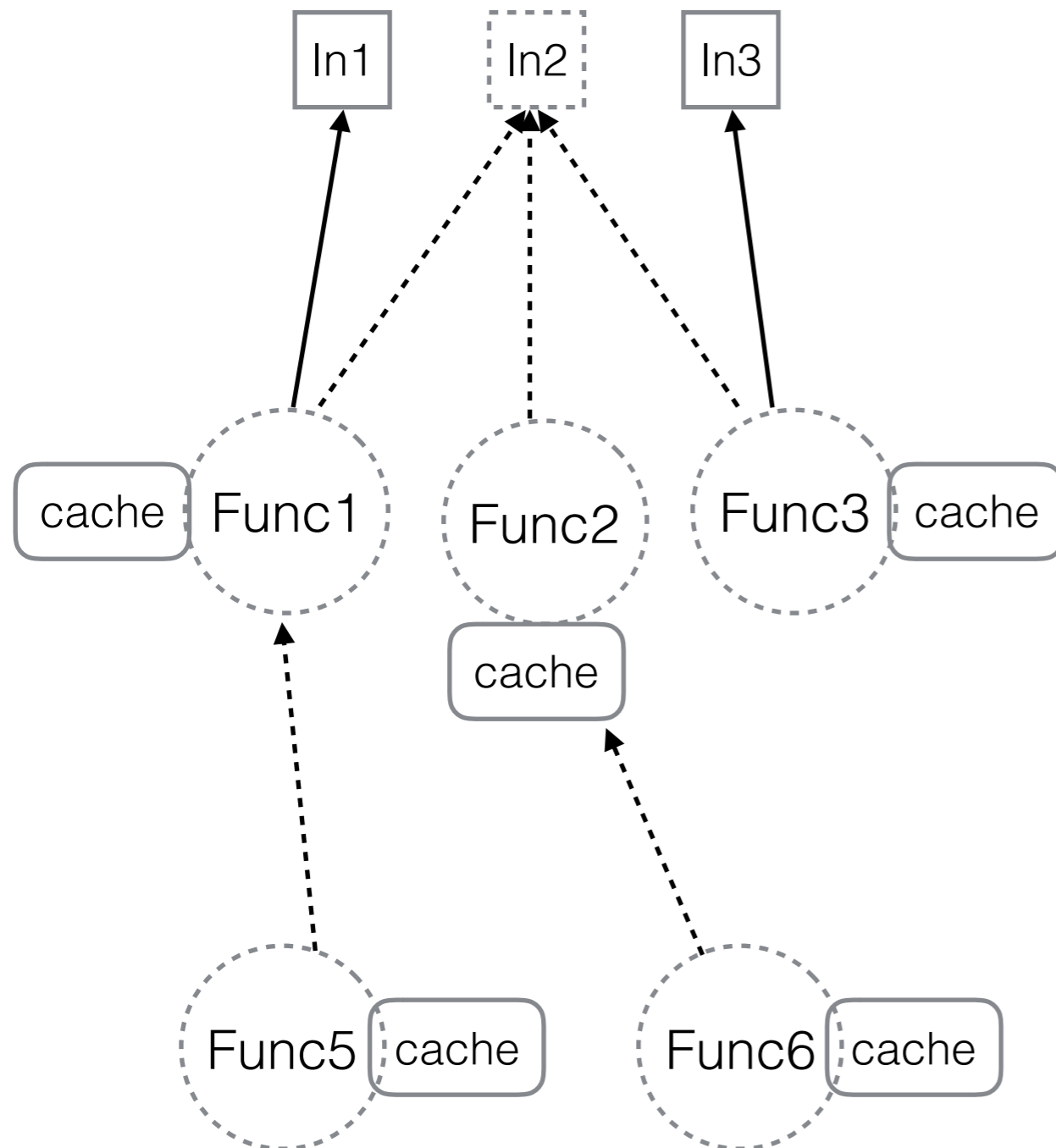
Minimum update paths

From-scratch consistent

Dynamic call-graph

Adapton

Build and maintain a partial-order call graph of cached results



On-demand re-evaluation

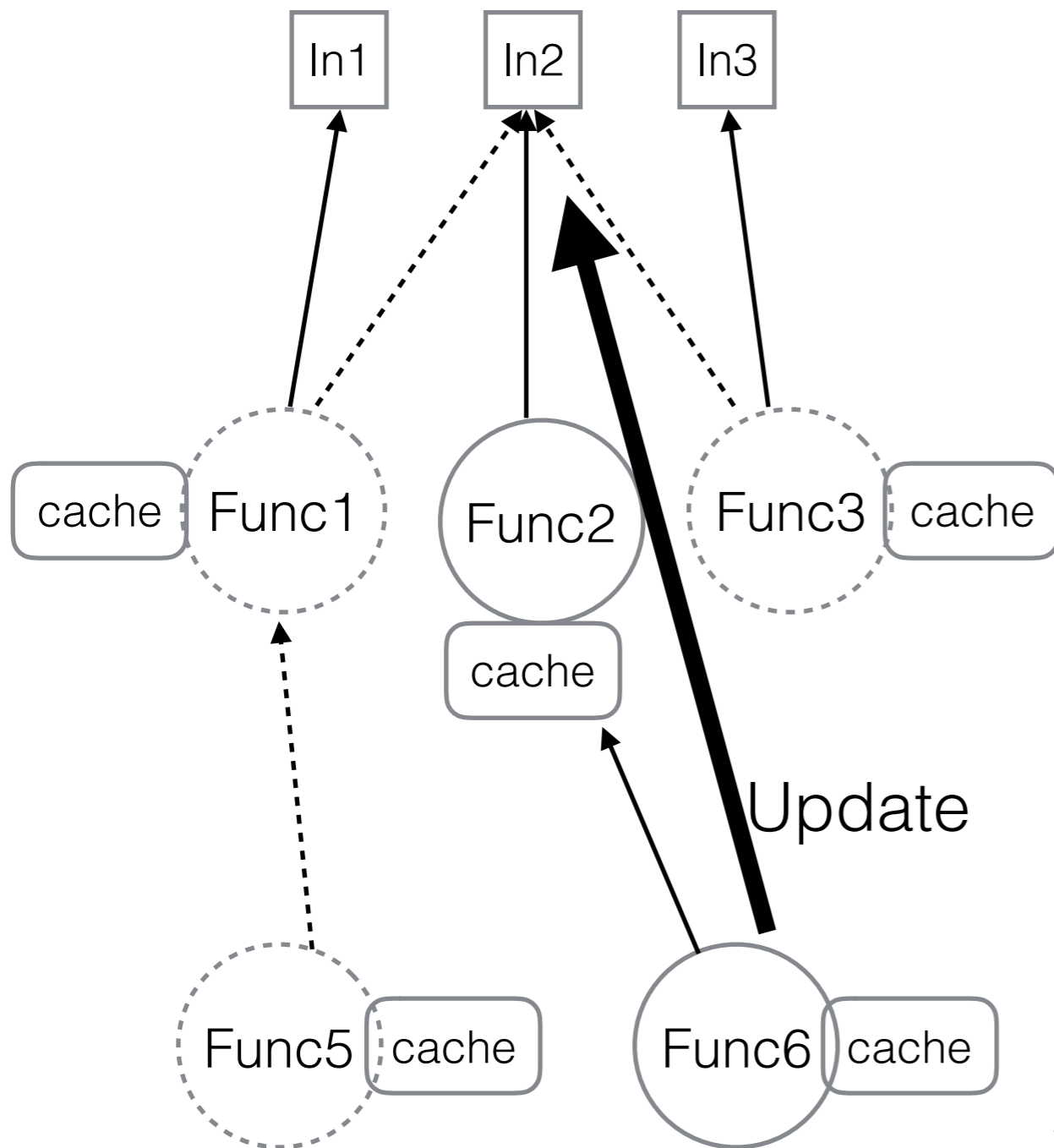
Minimum update paths

From-scratch consistent

Dynamic call-graph

Adapton

Build and maintain a partial-order call graph of cached results



On-demand re-evaluation

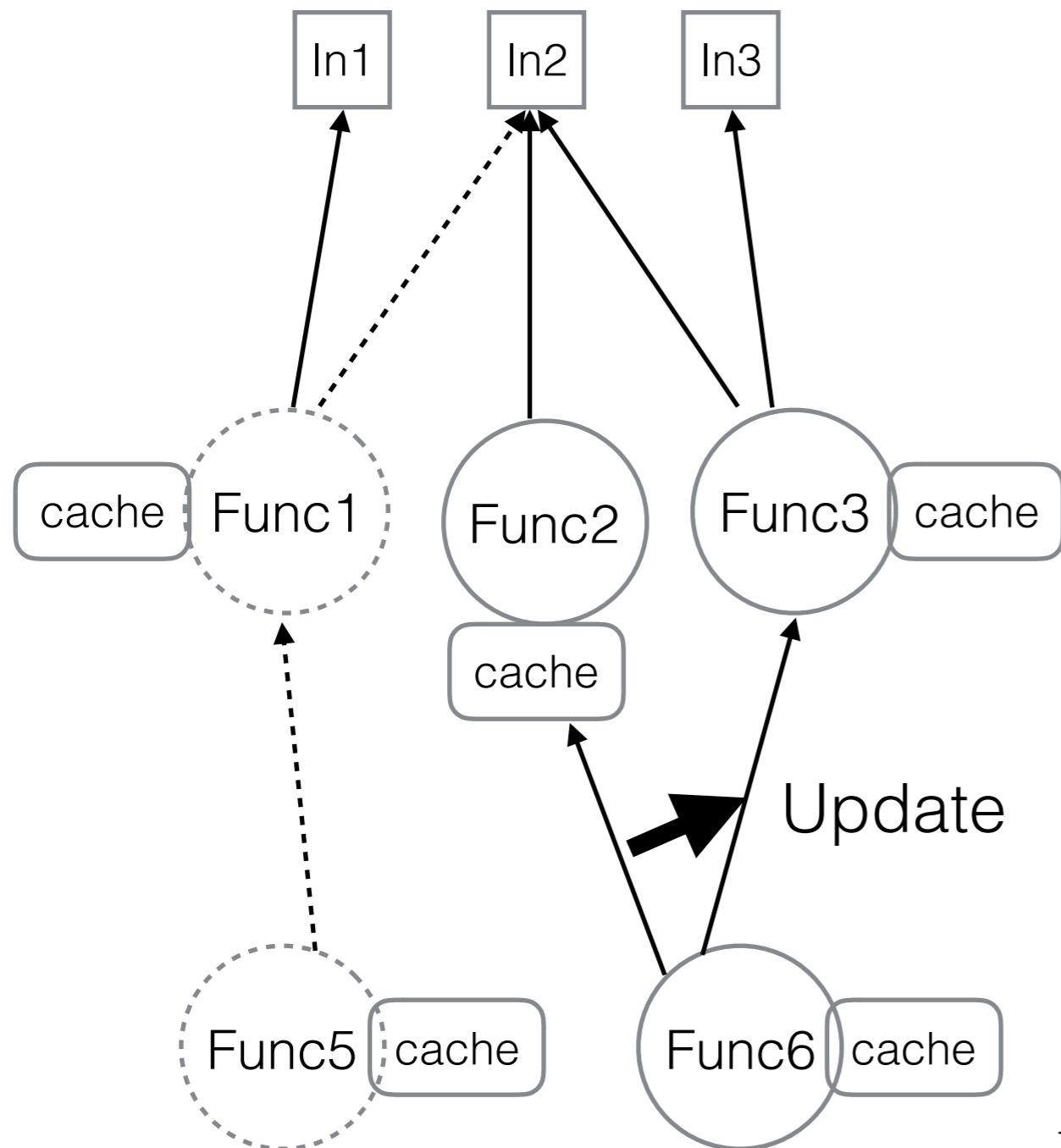
Minimum update paths

From-scratch consistent

Dynamic call-graph

Adapton

Build and maintain a partial-order call graph of cached results



On-demand re-evaluation

Minimum update paths

From-scratch consistent

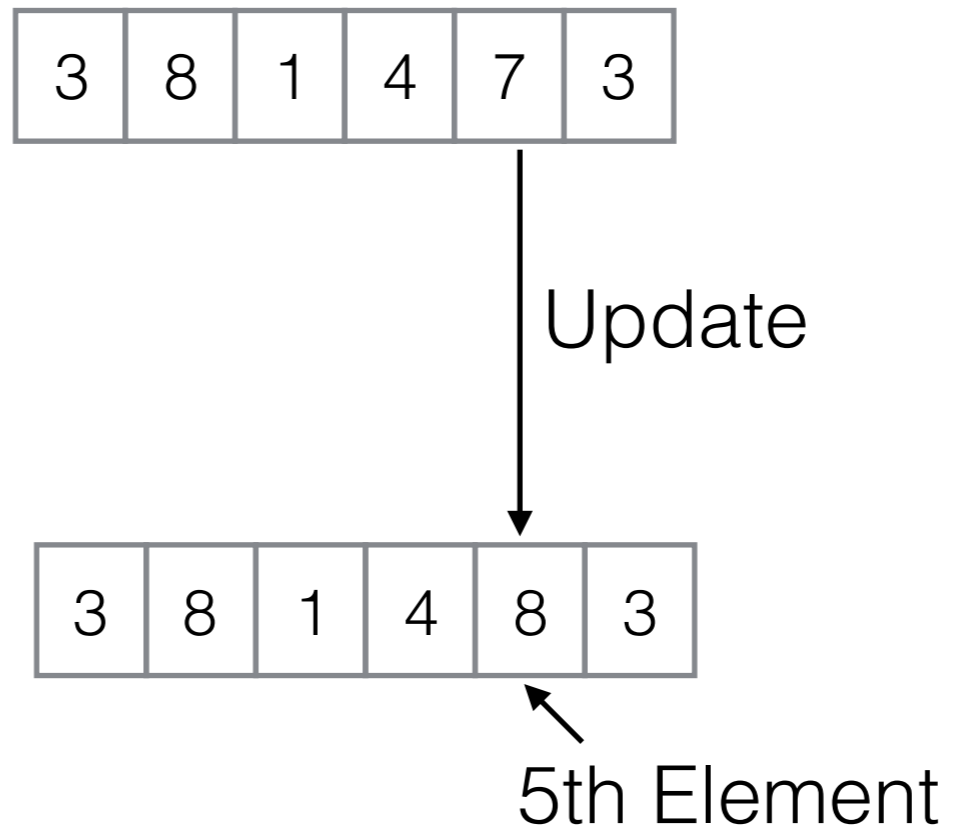
Dynamic call-graph

If Func2 and Func3 are the same function with different parameters, how should the cache and nodes be updated?

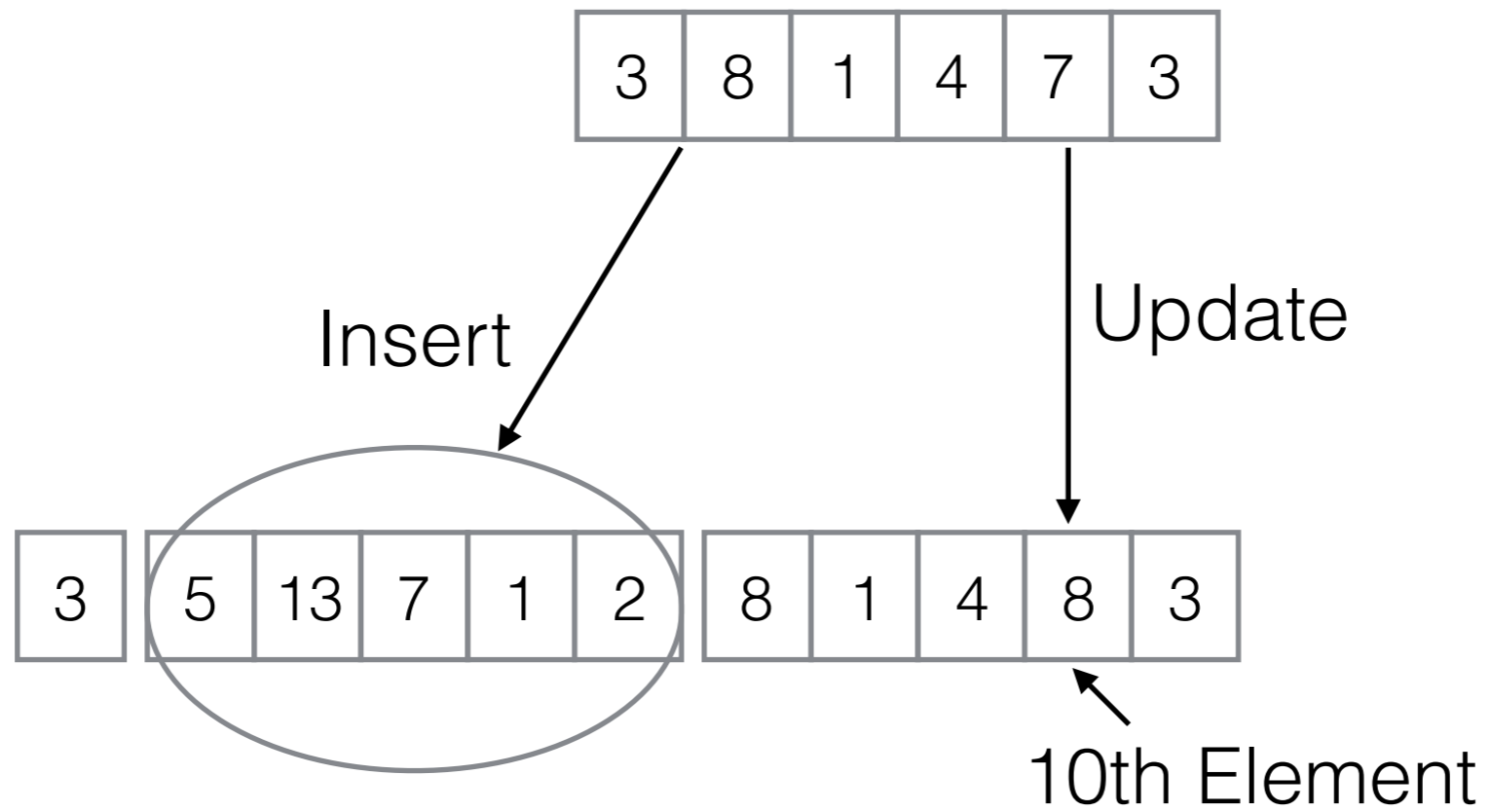
Incremental Sequences

3	8	1	4	7	3
---	---	---	---	---	---

Incremental Sequences

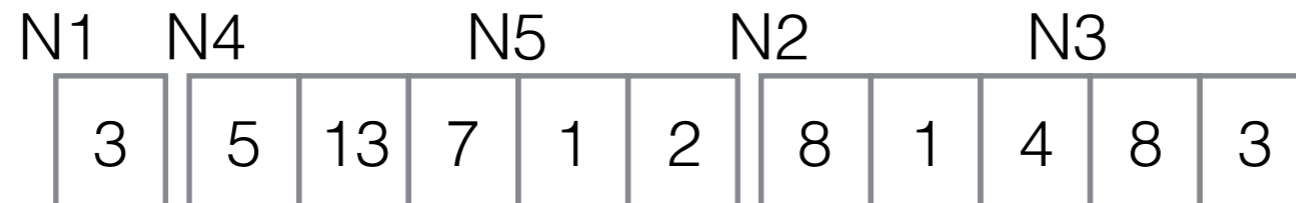


Incremental Sequences



Incremental Sequences

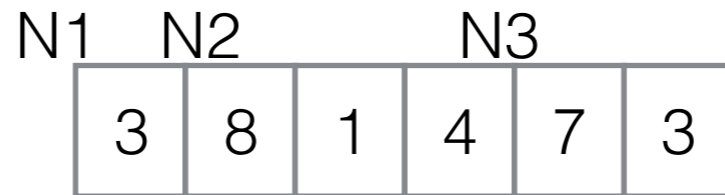
Use names to identify
specific sections



Incremental Sequences

Use names to identify
specific sections

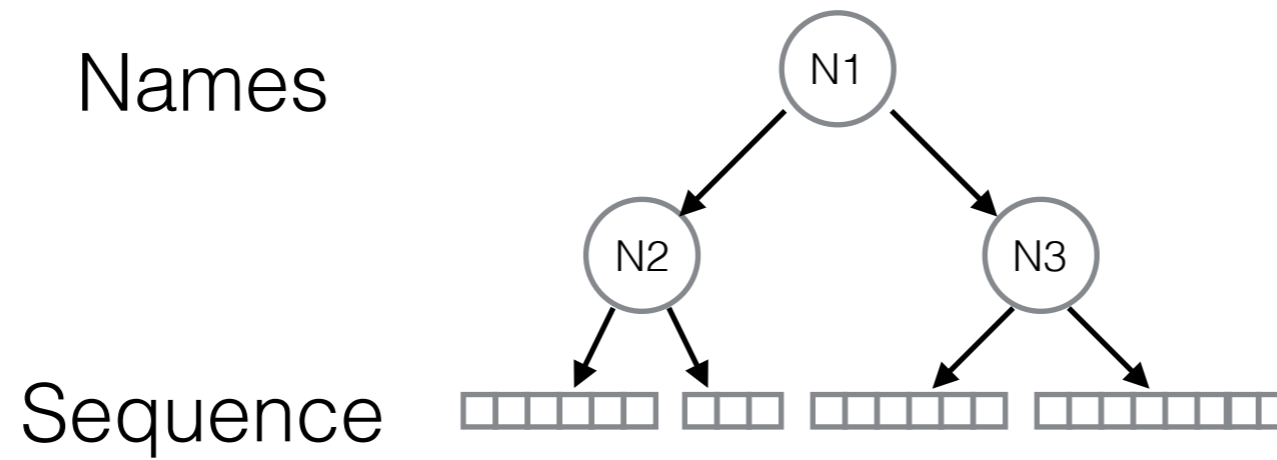
But how do you name
them, and how do
you avoid conflicts?



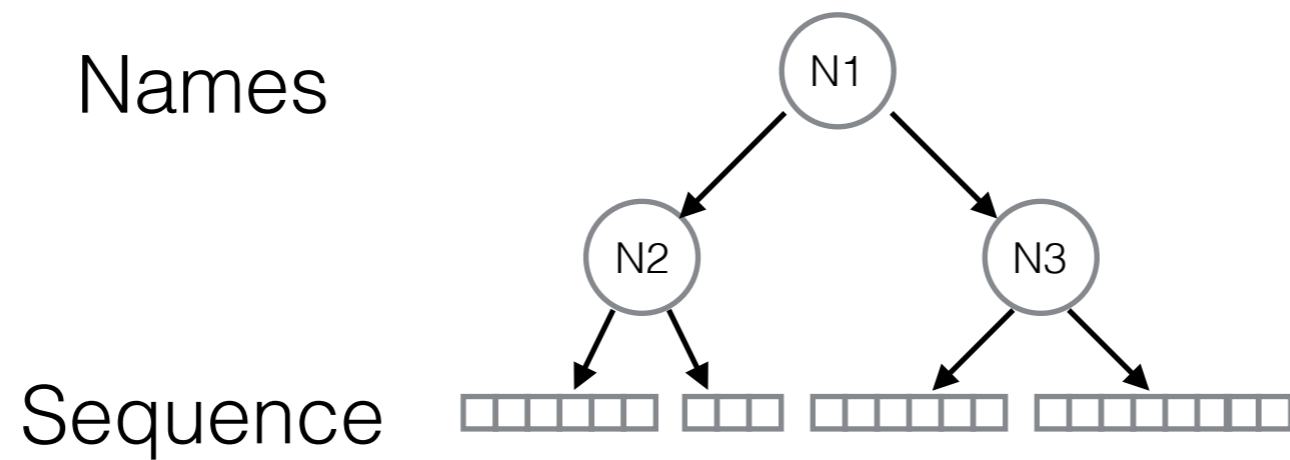
Giraz - Incremental Sequence

Sequence 

Giraz - Incremental Sequence



Giraz - Incremental Sequence



Named branches

Primitive arrays

Edit as zipper

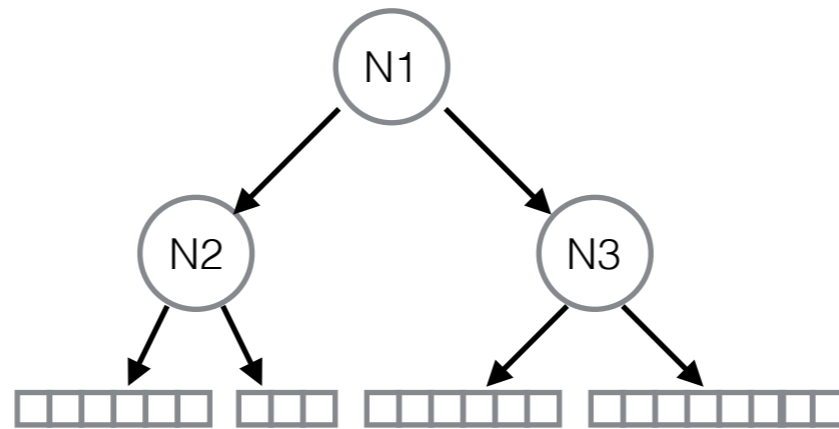
Canonical structure

Mimics Adaption structure

Giraz - Incremental Sequence

Names

Sequence



Named branches

Primitive arrays

Zipper



Cursor

Edit as zipper

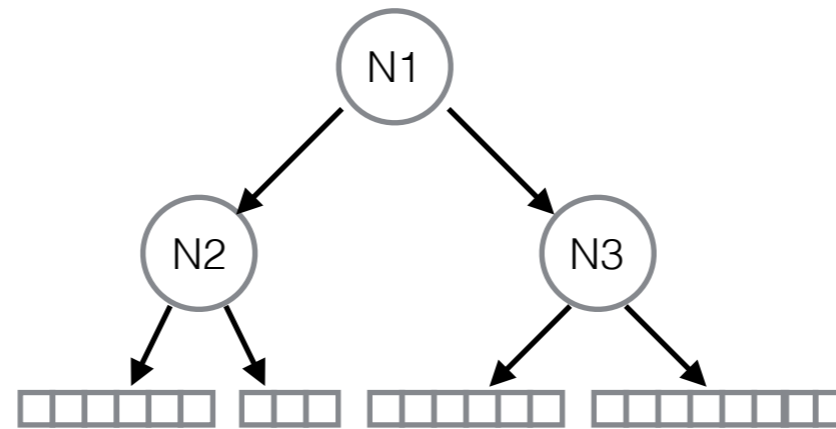
Canonical structure

Mimics Adaption structure

Giraz - Incremental Sequence

Names

Sequence



Named branches

Interface

- fold_lr
- fold_up
- map

Primitive arrays

Edit as zipper

Canonical structure

Mimics Adaption structure

Giraz - Incremental Sequence

	speedup over native Rust	Native initial run	Incremental initial run	Incremental update	crossover
to_string	449.0	93.80	95.50	0.21	1
reverse	22.2	2.01	7.85	0.09	4
max	57.5	2.84	5.99	0.05	2
adder	23.9	10.30	91.10	0.43	10
quickhull	4.2	56.60	213.00	13.50	6

Future work - IODyn

Design a language with implicit incremental features

Additional incremental data structures

- Tries
- Graphs

Annotations for expected incremental properties

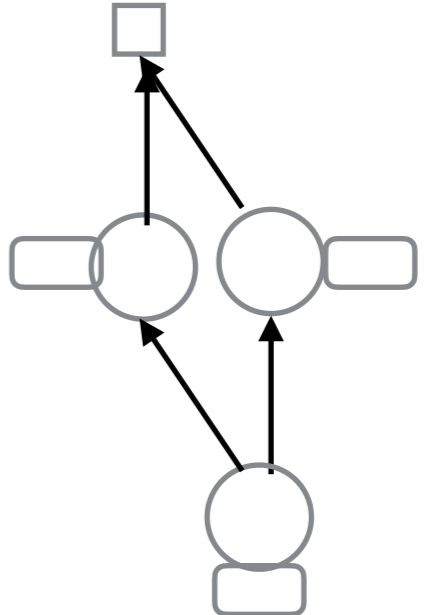
Type system to enforce proper name usage

Meta-proofs about consistency with from-scratch runs

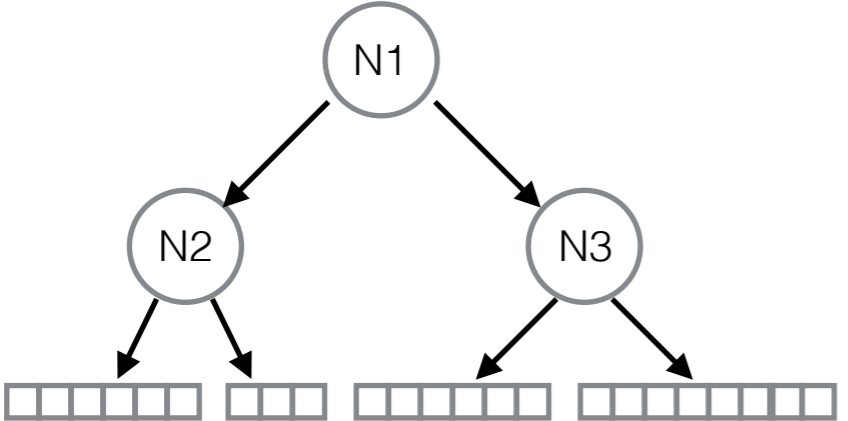
Summary

- Significant tools
- data structures
 - caching

Adapton dependency graphs



Giraz incremental sequence



Future:
Provide access to incremental features usable in general purpose code