Understanding Garbage Collectors

in object-oriented programming

Lecture 2 - Generational Garbage Collectors

Nahuel Palumbo

nahuel.palumbo@inria.fr

CERCIRAS 2023

Memory Management

Garbage Collection

Automatic Memory Management

Garbage Collectors

Compute the size
Allocate in the memory

```
data = malloc(size)
...
... use data ...
...
free(data)
```

Manually
Memory Management

```
data = Data new
...
... use data ...
...
??????????
```

Maybe move the data for better use of the memory

Free the space when data is not used anymore

Automatic Memory Management

Garbage Collectors

Compute the size
Allocate in the memory

```
data = malloc(size)
...
... use data ...
...
free(data)
```

Manually
Memory Management

```
data = Data new
...
... use data ...
...
??????????
```

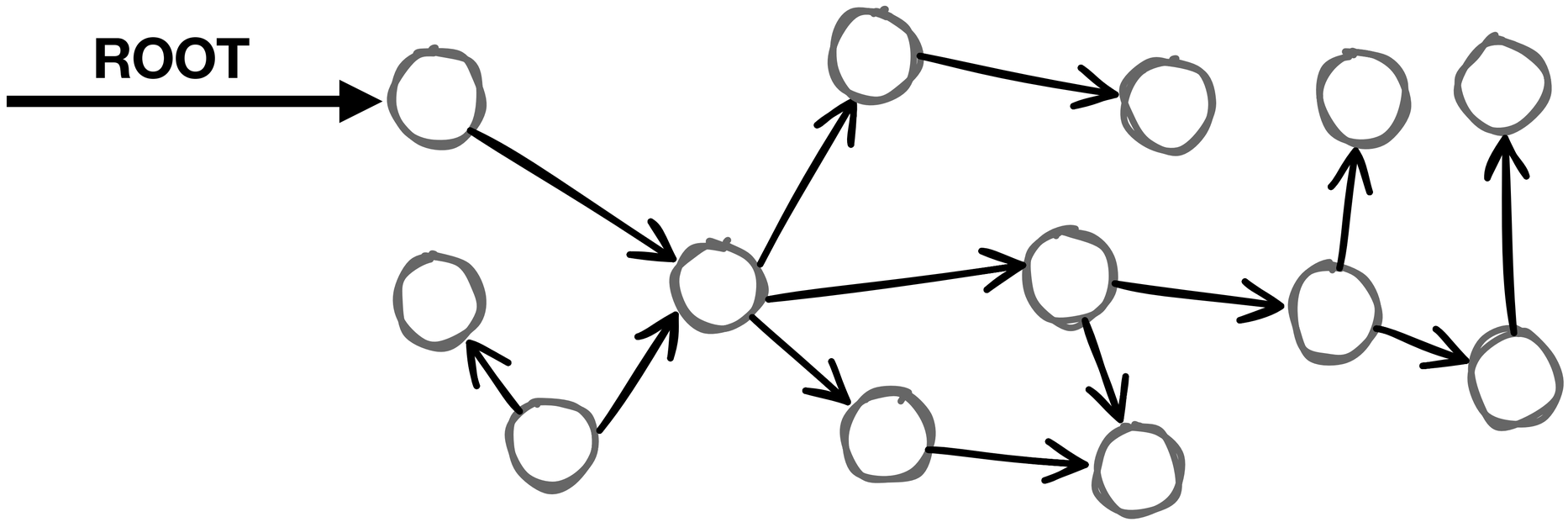
Maybe move the data for
better use of the memory

Free the space when data
is not used anymore

Developer

Application's Allocation Patterns

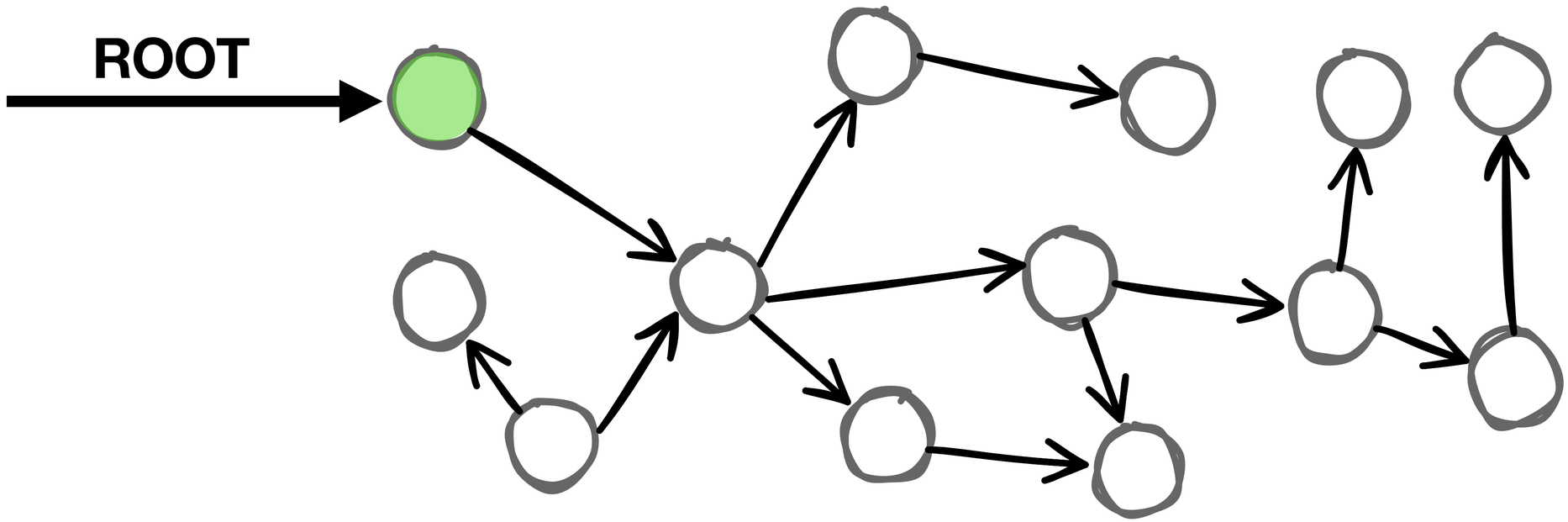
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

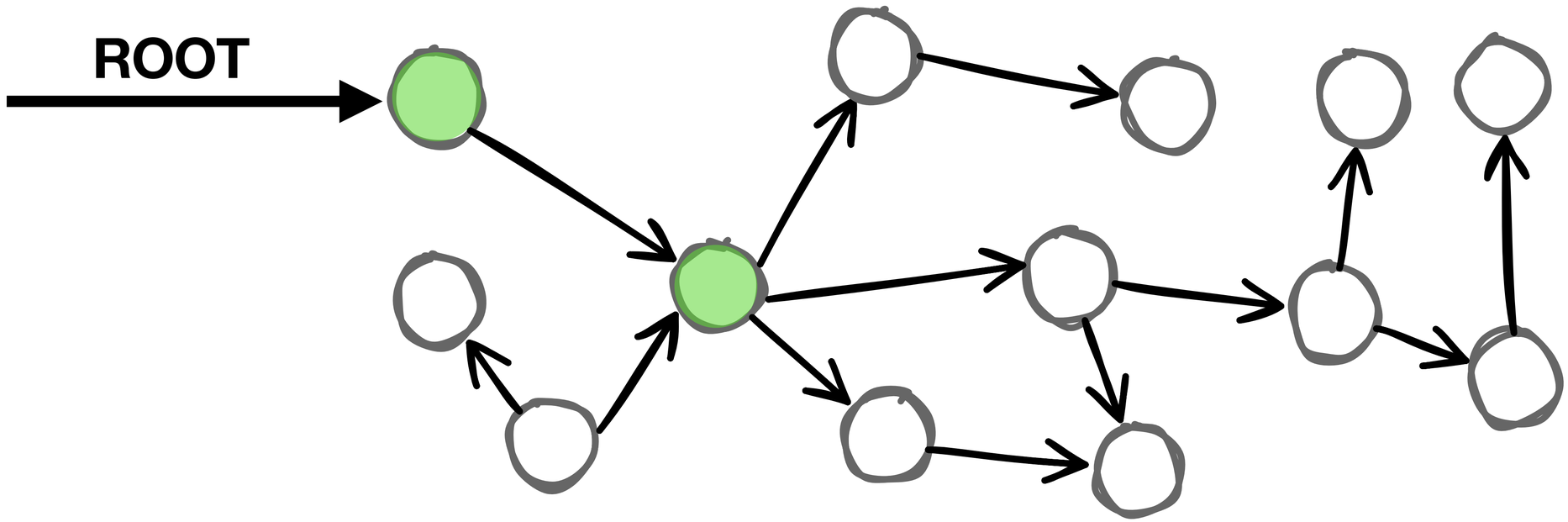
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

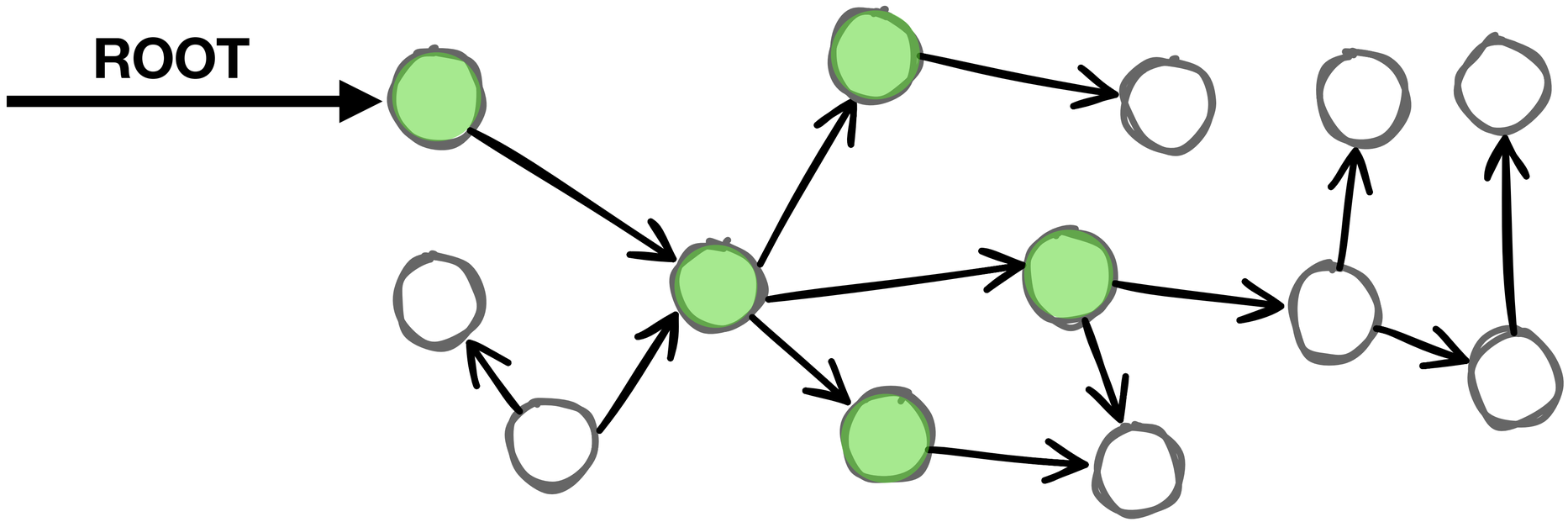
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

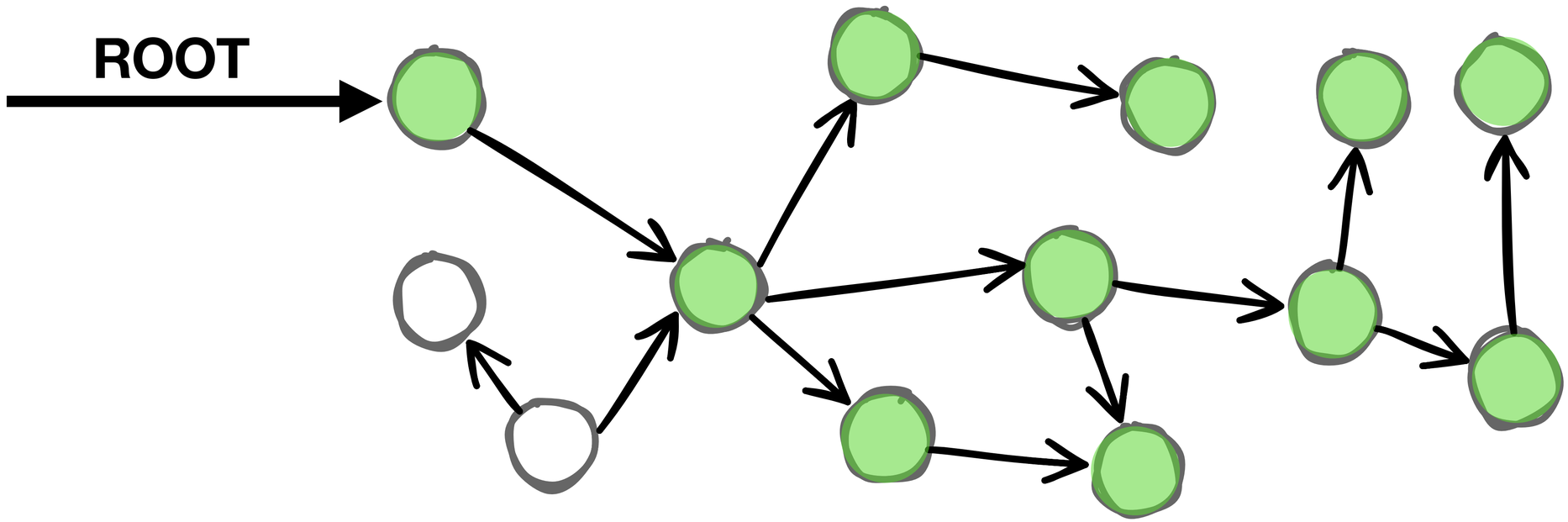
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

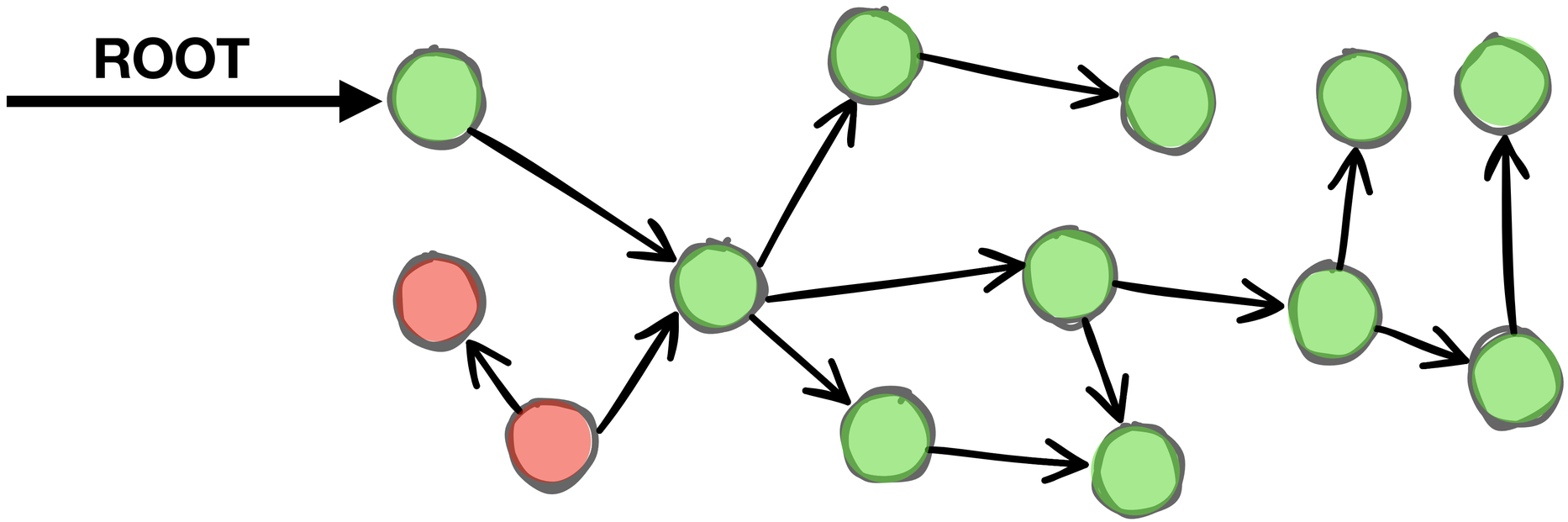
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

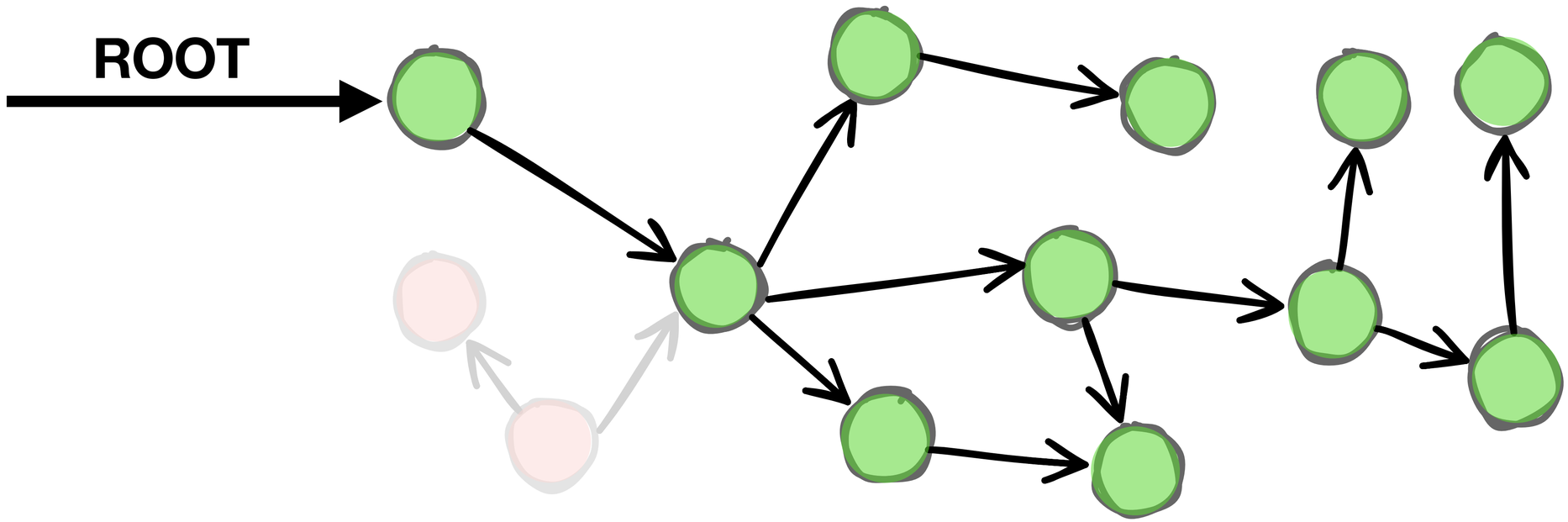
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

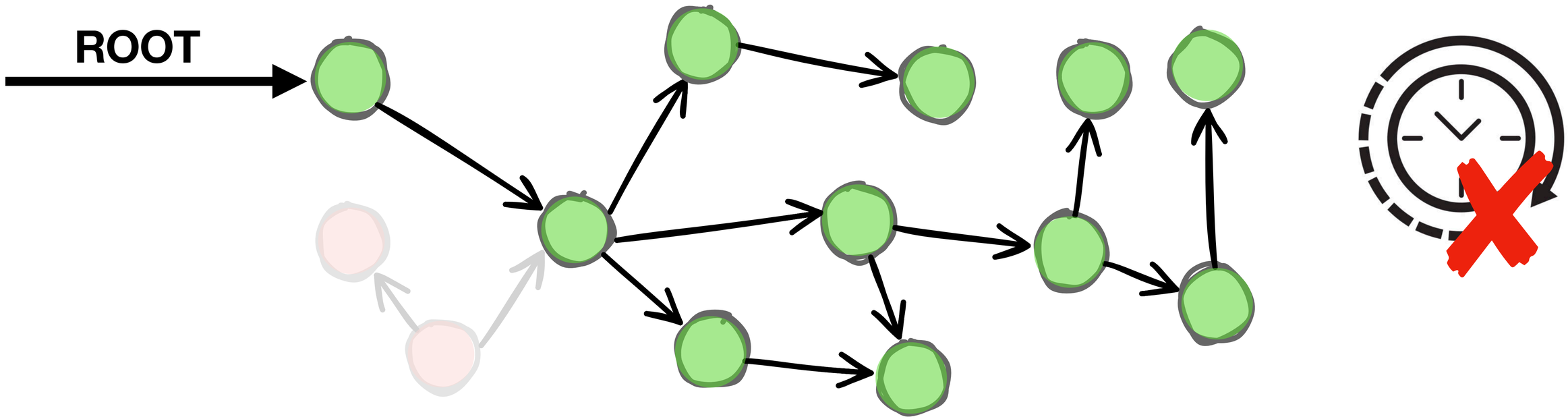
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

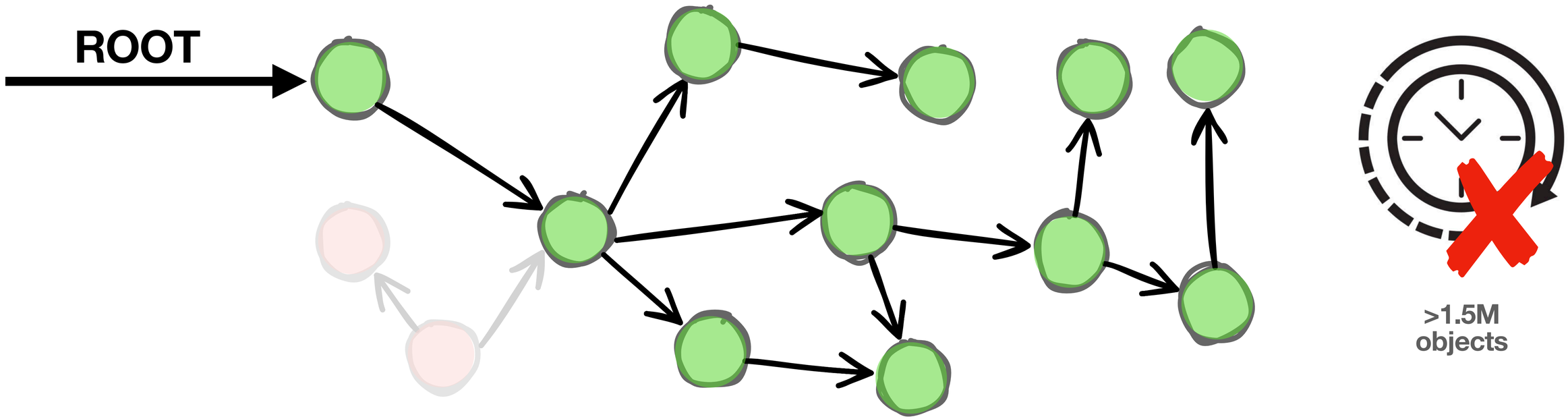
When an object dies?



"Must be accessible from the roots"

Application's Allocation Patterns

When an object dies?



"Must be accessible from the roots"

Automatic Memory Management

Garbage Collectors

Compute the size
Allocate in the memory

```
data = malloc(size)
...
... use data ...
...
free(data)
```

Manually
Memory Management

```
data = Data new
...
... use data ...
...
??????????
```

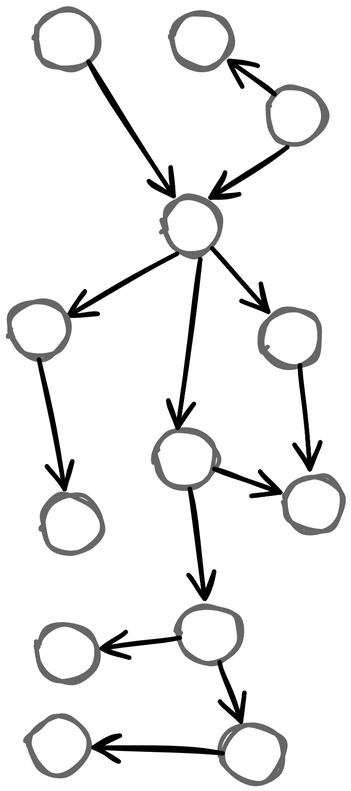
Maybe move the data for better use of the memory

Developer

Free the space when data is not used anymore

Application's Allocation Patterns

How do the applications use the memory?

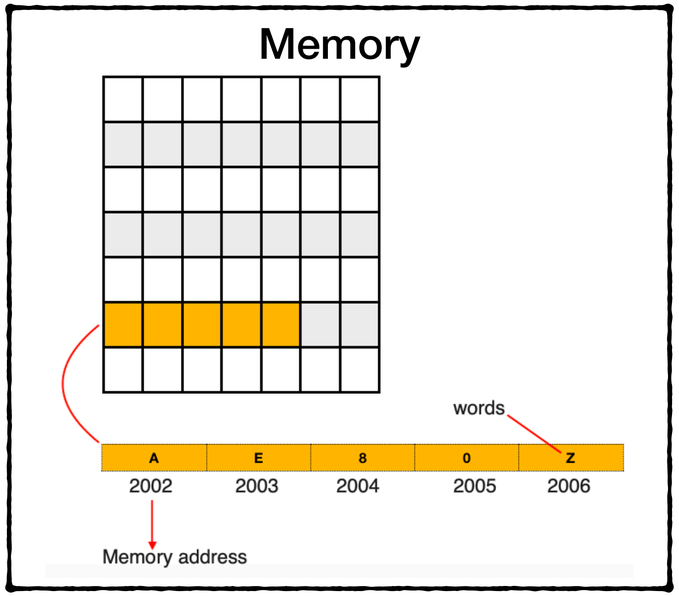


Allocations are particular for each application

→ **Hard to predict**

There are some general heuristics

→ **Easy to predict**



Application's Allocation Patterns

Weak generational hypothesis

```
data = Data new
... first use of data ...
??????????
```

Will you use it again?

Mostly

No, you can free the memory

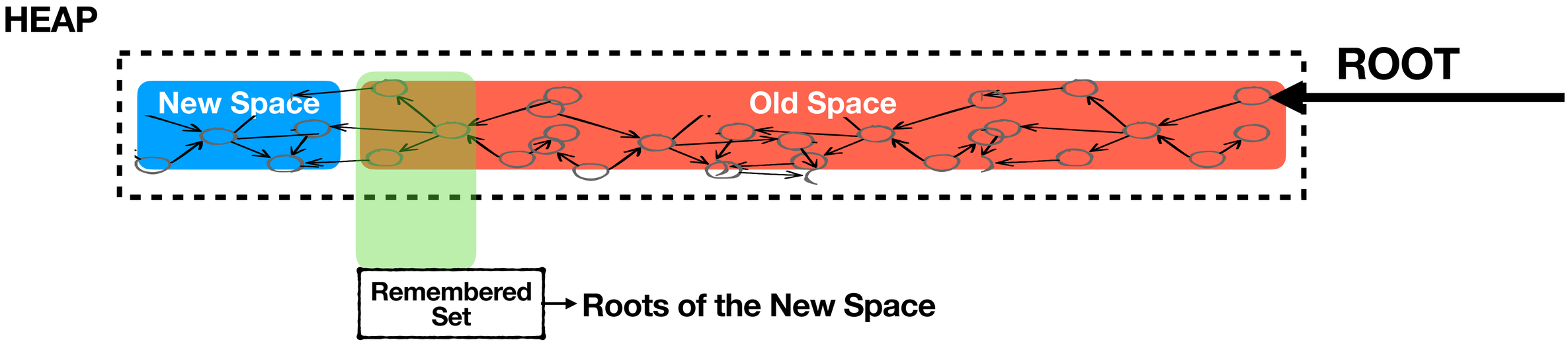
Sometimes

Yes, keep it!

"Most of the objects die young"

Generational Garbage Collector

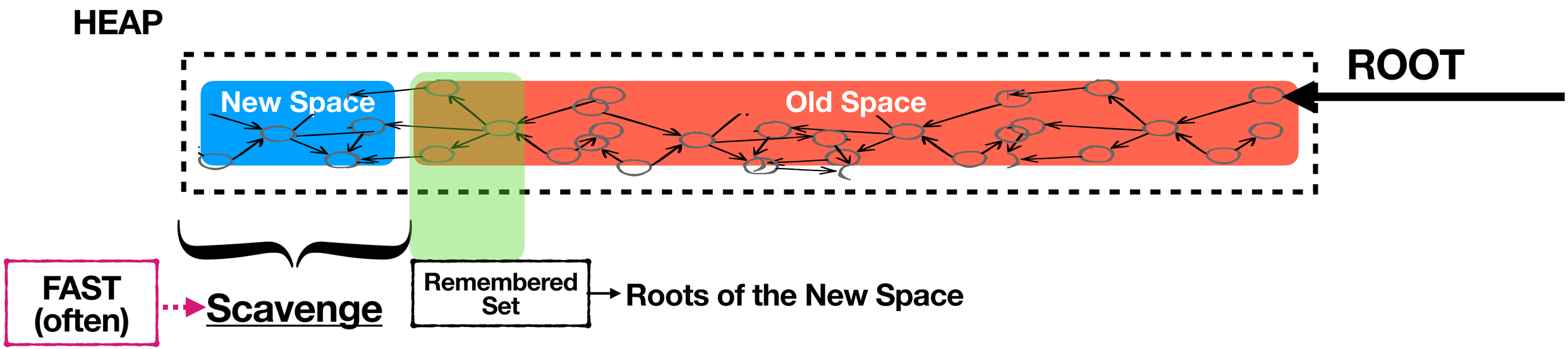
High-Performance Automatic Memory Management for OOP



D. Ungar, *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*

Generational Garbage Collector

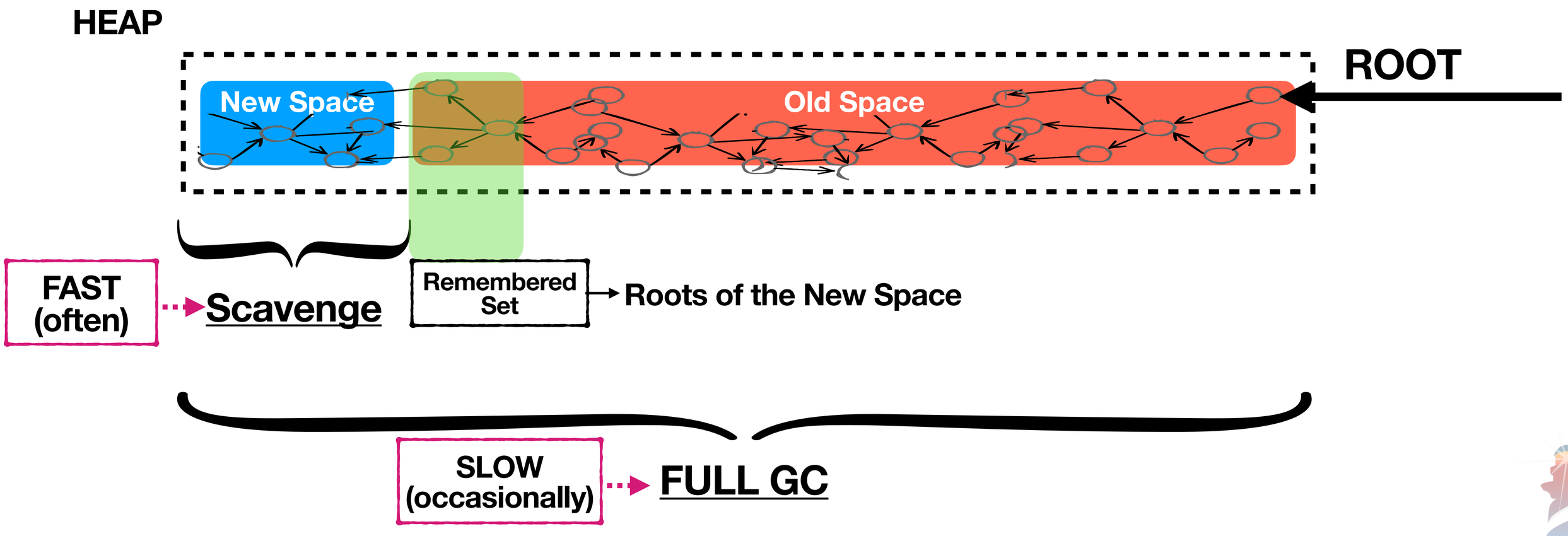
High-Performance Automatic Memory Management for OOP



D. Ungar, *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*

Generational Garbage Collector

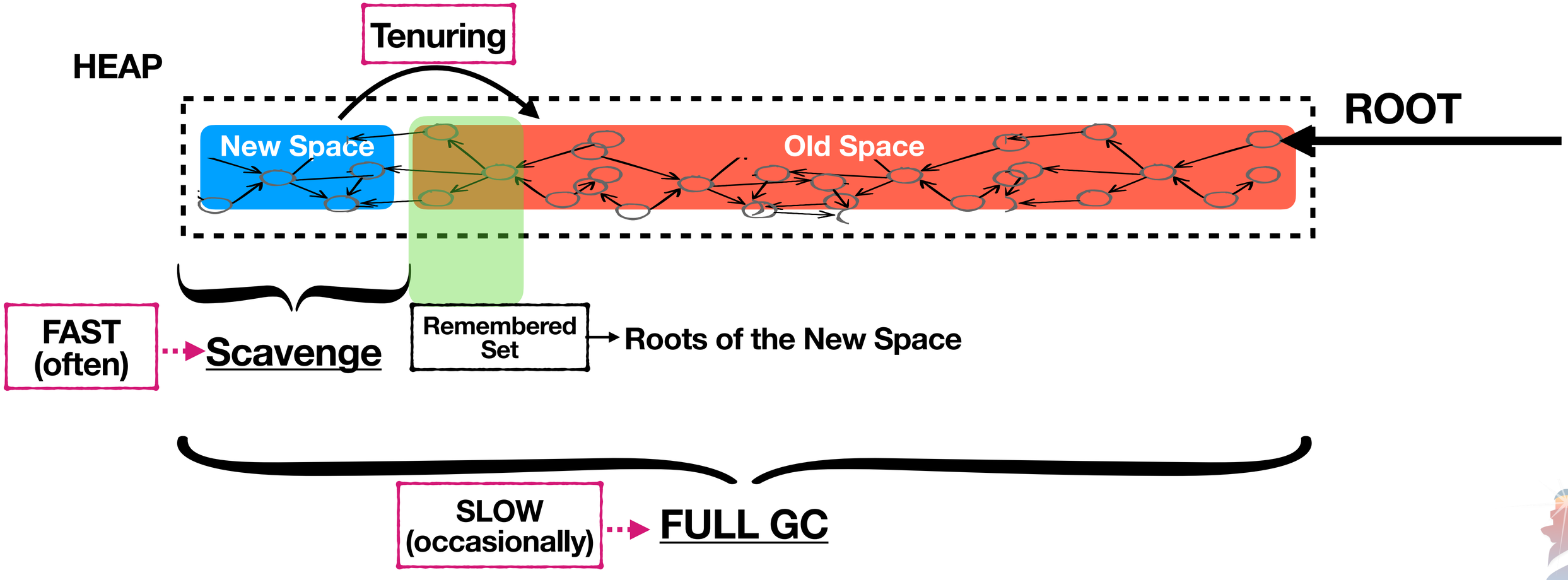
High-Performance Automatic Memory Management for OOP



D. Ungar, *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*

Generational Garbage Collector

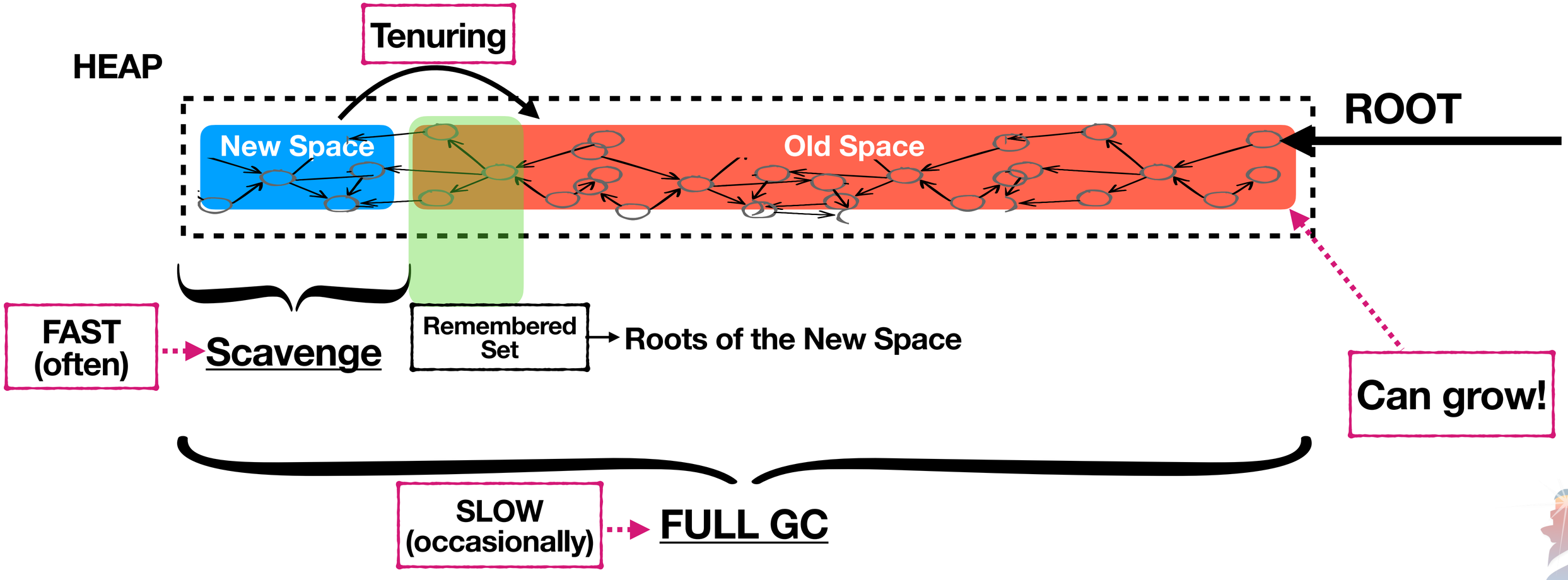
High-Performance Automatic Memory Management for OOP



D. Ungar, *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*

Generational Garbage Collector

High-Performance Automatic Memory Management for OOP



D. Ungar, *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*