Challenge D

Challenge Goals

Replaces Cheney's algorithm with Mark-and-Sweep algorithm

Approach in Lecture 6 not sufficient for dynamic node sizes.

Cheney's Algorithm

Splits memory space into two equal halves

Copying of roots to next half

Breadth-first copying of children nodes

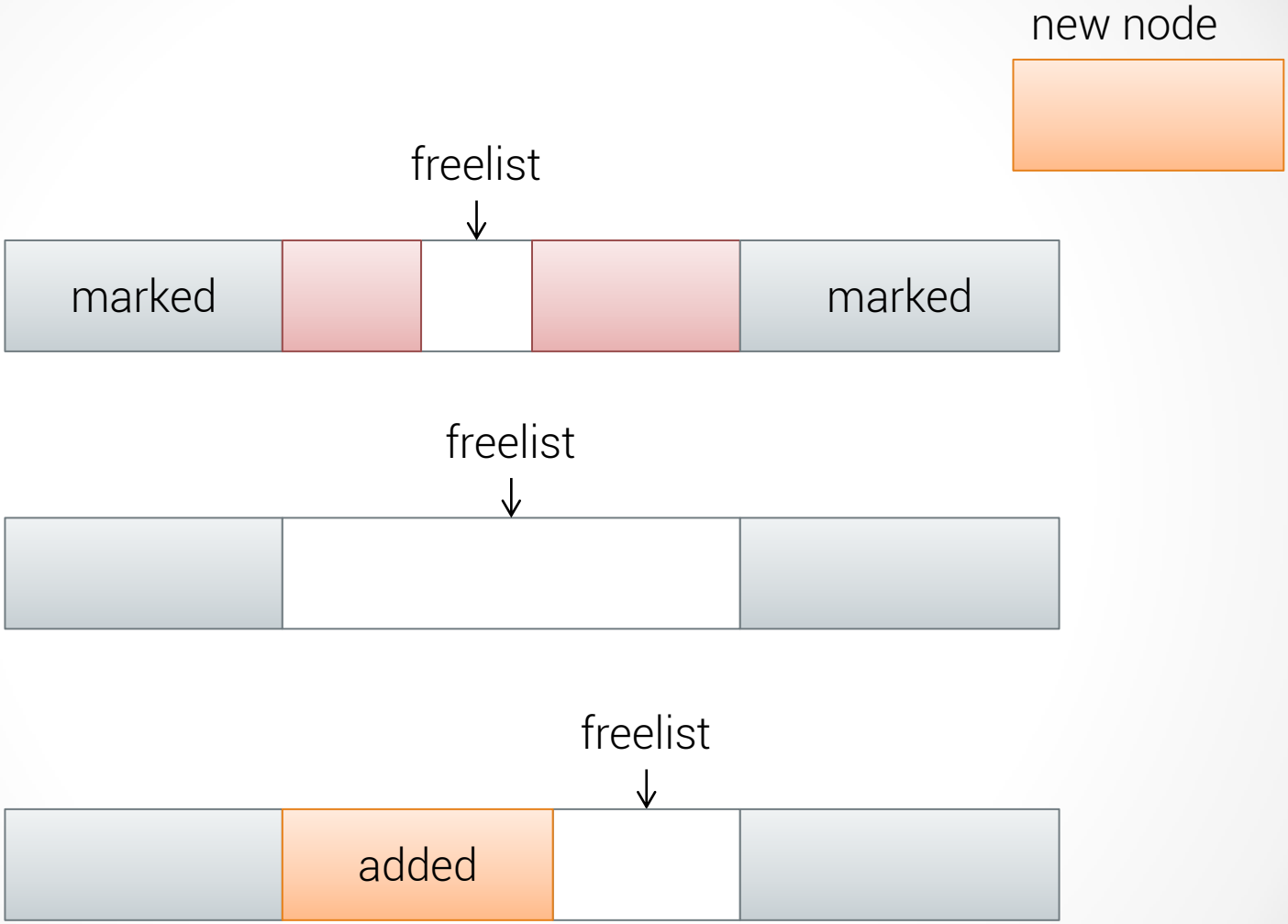
Mark-and-Sweep Algorithm

Uses free list to track free spaces

Depth-first marking of live nodes

Sequential recovery of free spaces

Mark-and-Sweep Algorithm



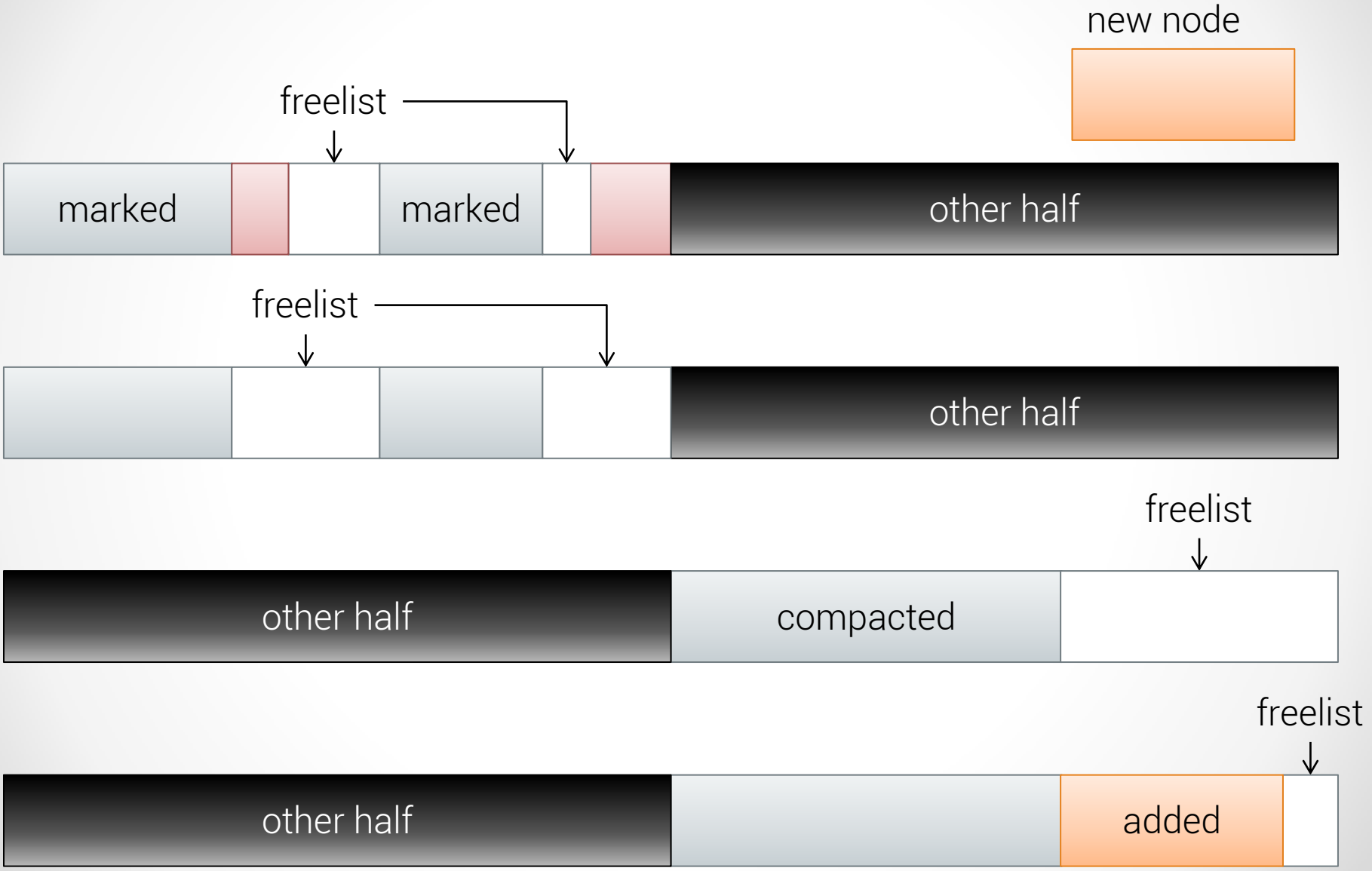
My Solution

Splits memory space into two equal halves

Performs Mark-and-Sweep algorithm first

Performs Cheney's algorithm for compaction upon lack of available space

My Solution


new node
freelist
marked
marked
other half
freelist
other half
freelist
other half
compacted
freelist
other half
added

Code Demonstration

Generational Garbage Collection

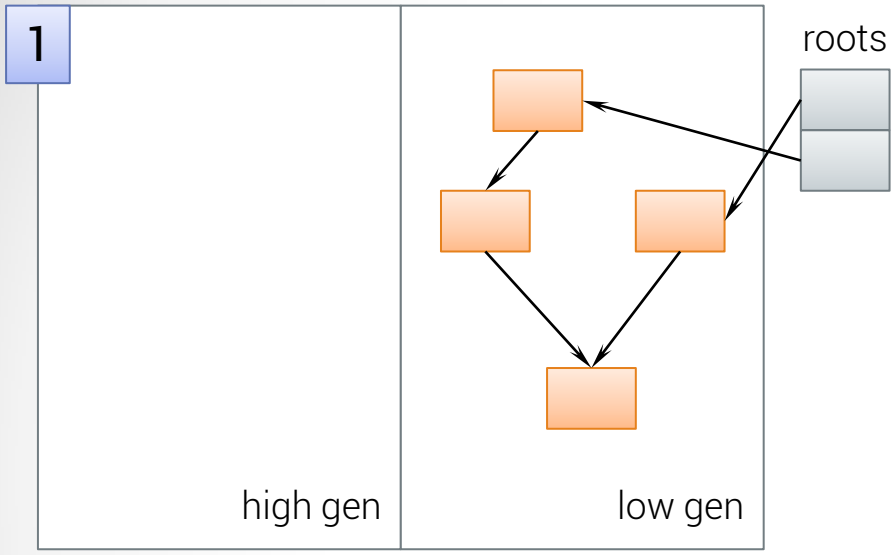
Definition

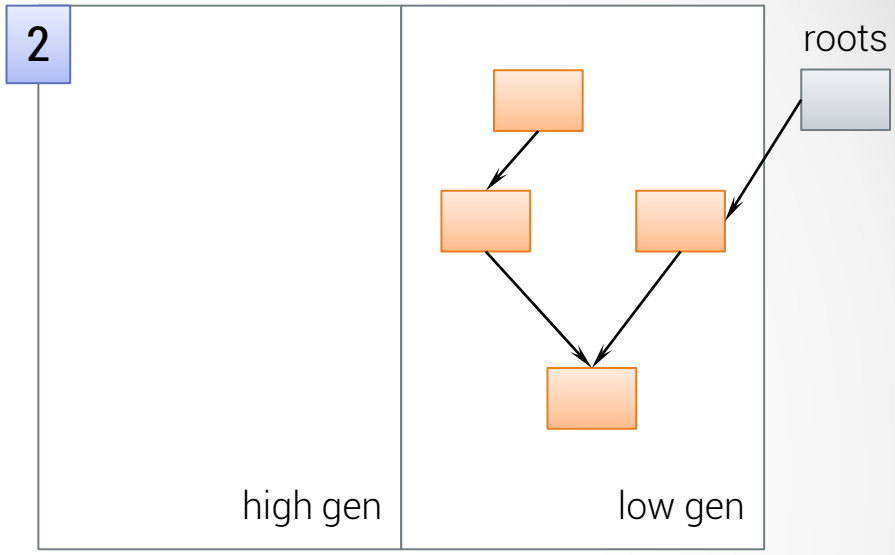
Generational GC divides and allocates nodes into different generations

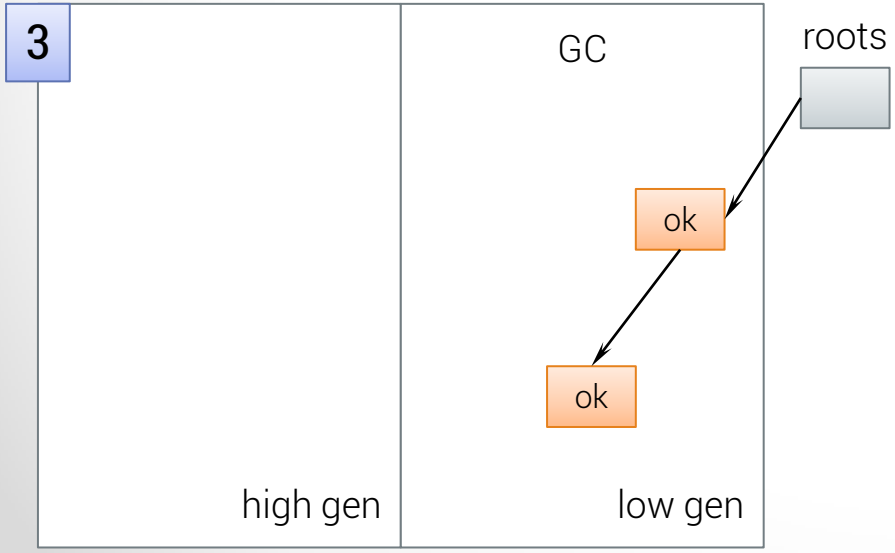
New nodes are created on lowest generation level

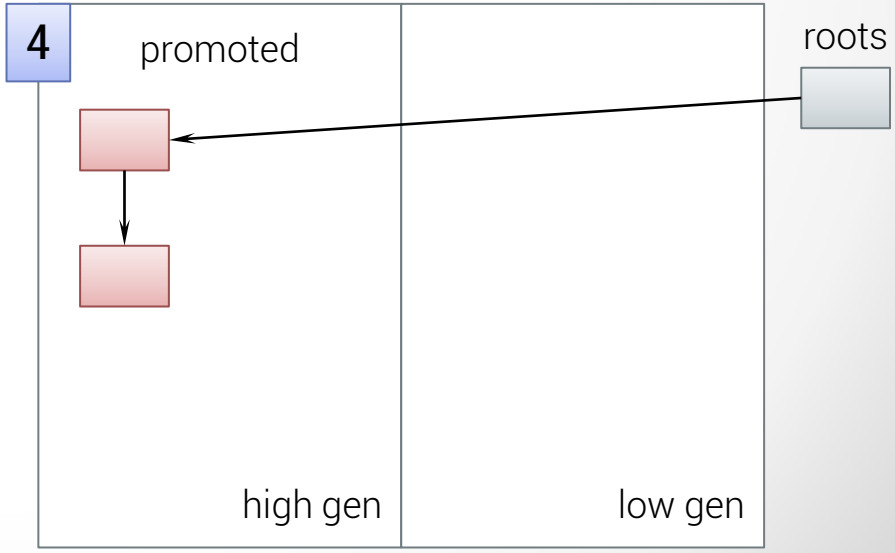
Nodes surviving the GC will be promoted to higher generation level

Understanding

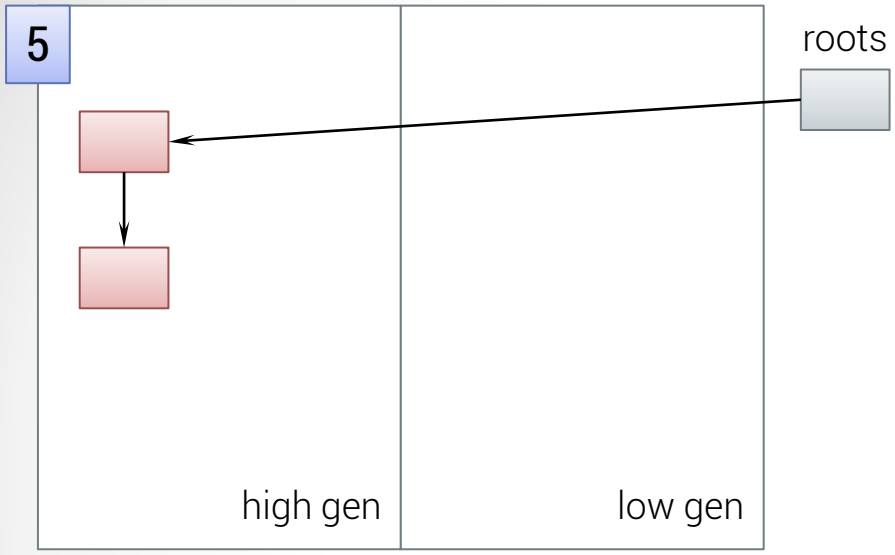


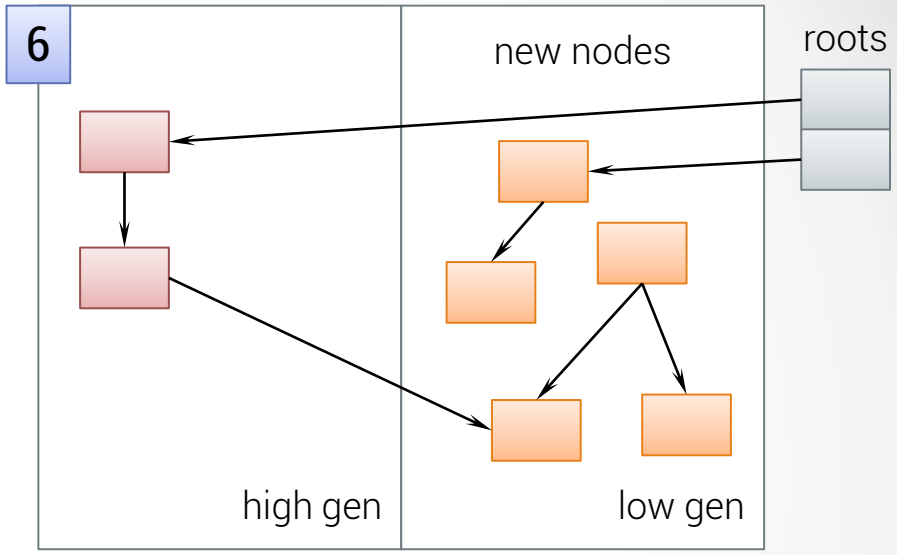


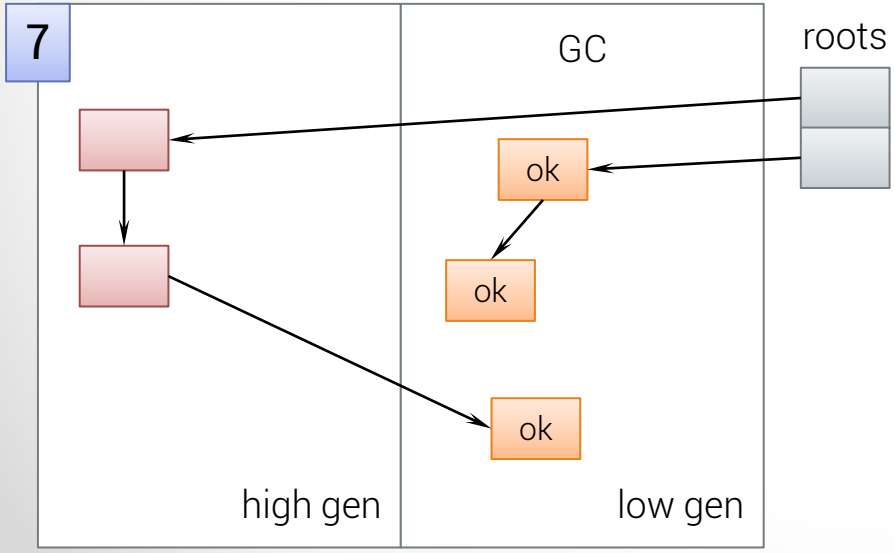


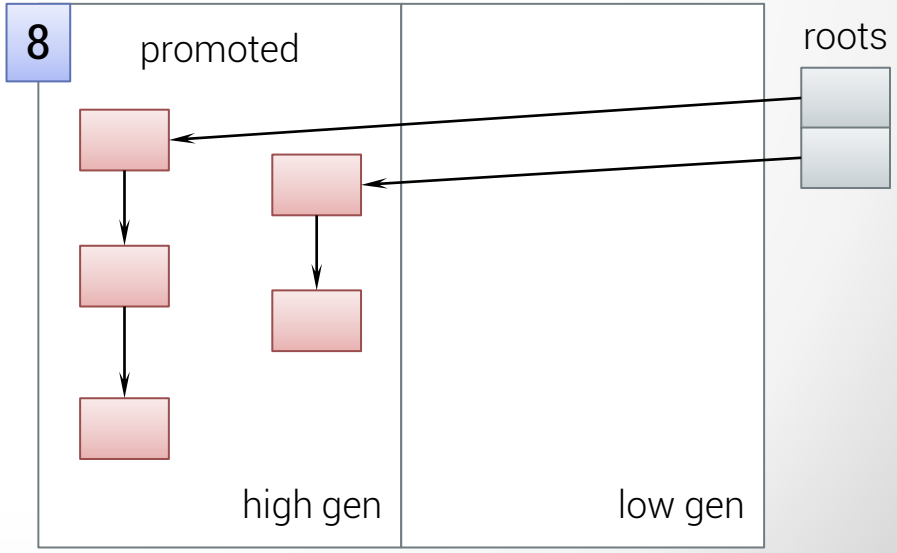


Understanding









Understanding

Garbage collection starts with the lowest generation

Generally only requires two levels of generation to be effective

Smaller heap for lower generation, (much) larger heap for higher generation

Understanding

Very large proportion of nodes at lowest generation can be garbage collected

Heuristic approach that permits a faster approach of GC

Improvements to Mark-and-Sweep

Mark-and-Sweep algorithm is very sensitive to size of memory space

Much smaller memory space outweighs the lower residency disadvantage

Effectively targets a localized memory region where most nodes can be freed

Real Life Examples

Java

.NET