Reference Counting — Apple

DESIGN a Garbage Collector :-

USE CASES :-

1. Free / Reclaim memory from dangling objects / pointers
2. Periodic cycles / intervals / memory reached limit run GC without halting Program
3. Avoid memory leaks

High level design :-

1. Garbage Collector
   - tracing GC (Algo)
     - naive Mark & Sweep
     - Tricolor marking
   - Reference Counting GC (Algo)
     - Weighted Reference Counting
     - indirect reference counting

CONSTRAINT :-

1. Tracing GC : cycles with reclaim will halt the application
2. Mark & Sweep Algo :- Mutiple times access to same memory set. Unnecessary Visiting the memory set.

3. Reference Counting :- for reference pointing to each other ⇒ cyclic references.

Implementation of Garbage Collector :-

1. TRACING Garbage Collector :-

(A) Naive Mark and Sweep :-

Collector Cycle

Mark the reference objects

Next ( ) after marking

Sweep the non-reference objects

* Twice Visiting the memory set.

(B) Tri color Marking :-

Grey —— All objects

Black —— object with references but no reference to white area

White —— dangling pointers for recycle

when grey is empty from pushing to black the white is recycled

prevents twice visiting the memory

2. Reference Counting :-

1. Zero reference means recycle
   counter for each reference for each
   object
2. delete / reduce counter when references
   deleted
3. disadvantage :- We can have
   cycle, a pointer pointing to
   itself
   So we used storage modifiers
   (A) weak
   (B) Strong
   (C)
   IOS uses reference counting
   Garbage collection algorithm