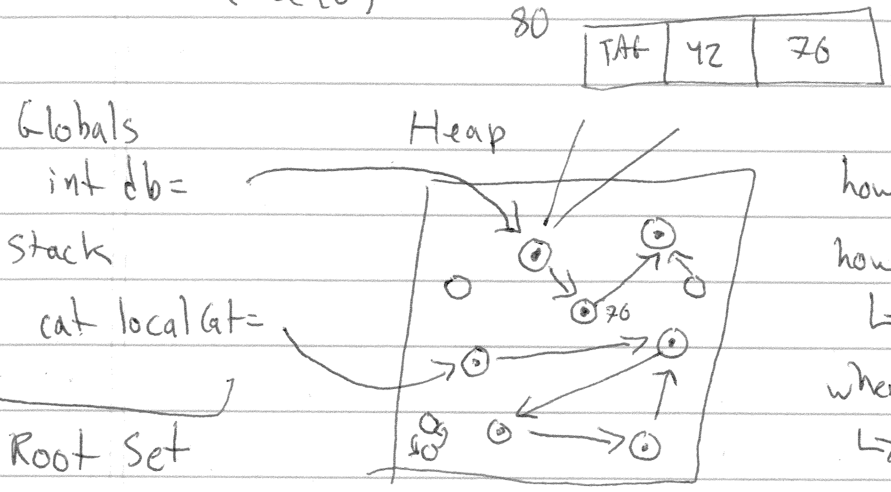
14-2 Mark + Sweep

"Garbage Collection"

↳ a correct memory manager

sound as complete as can be

```
for o in Object-Graph
    mark (o)
for o in memory
    if !marked (o)
        free (o)
```



how does GC know the roots?

how does GC know obj layout?

↳ tag, BiBoP

where is the mark?

↳ tag or bitmask per page

time → malloc is $O(\lg n)$ / free is $O(n)$ vs $O(n \lg n)$

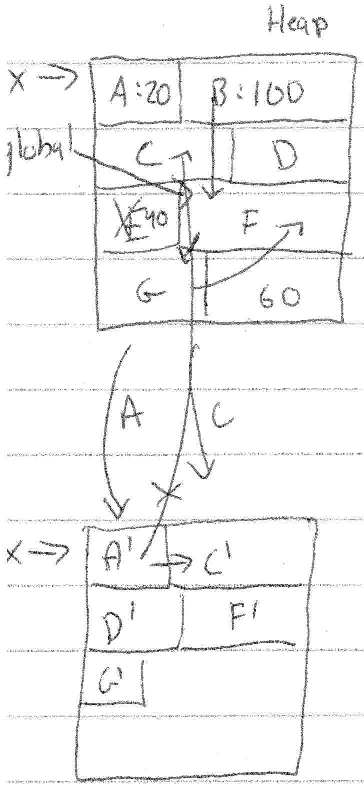
mem → tags

time → mark is $O(\text{live})$ / sweep $O(\text{memory})$

sound + complete / as can be

Hans Böhm GC for C

2-1 | Stop + Copy



mem ⇒ 2x overhead

malloc → O(1)

GC ⇒ O(live)

arr[27]

= *(arr+27)

can't alloc 100

maybe move F & G over

L ! F may be big

L ! what if F is a ptr target

Table of

src to dest

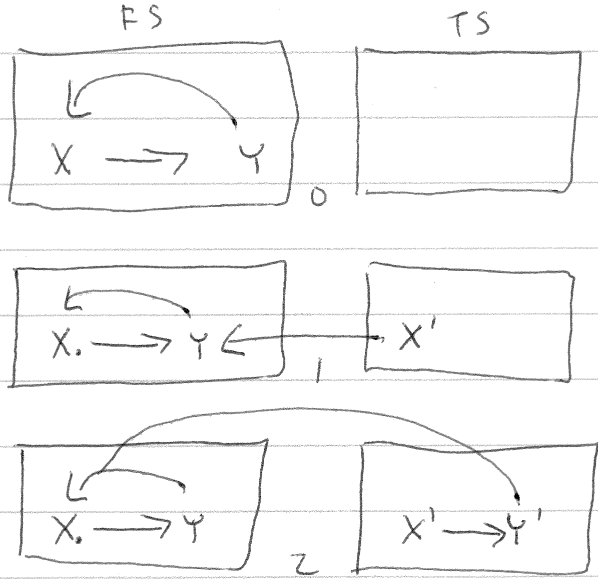
ptrs

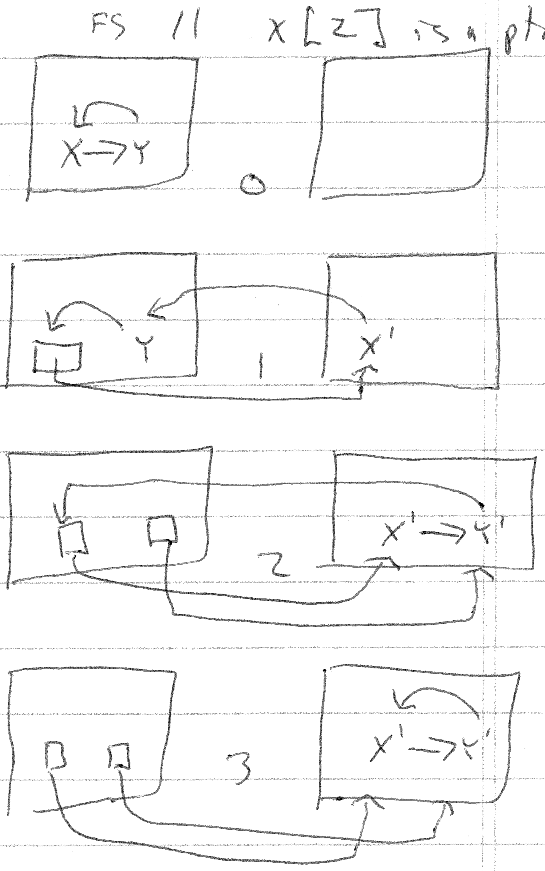
global or per-obj

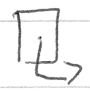
copy from FROMSPACE

to TOSPACE

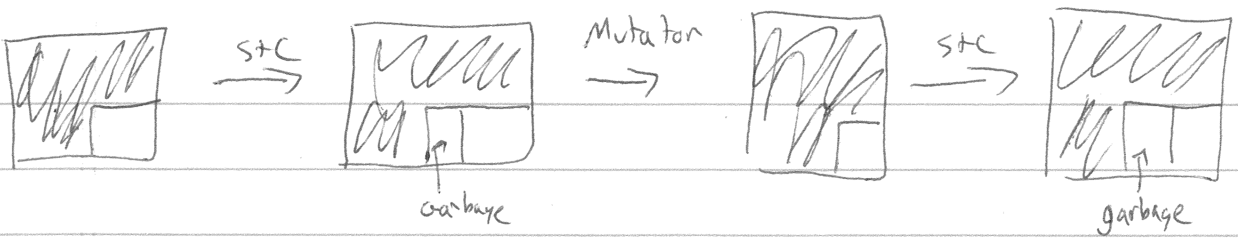
```
for o in root set or Q          // global *x = 100
    copy o (shallow) to TS      // copy 100 ... 105
    updateref to o              //   to 300 ... 305
    add o's refs to Q           // x = 300
                                // x[2] is a ptr to 50
```

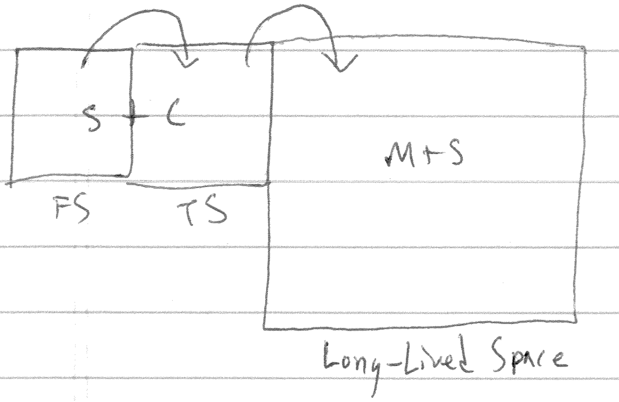


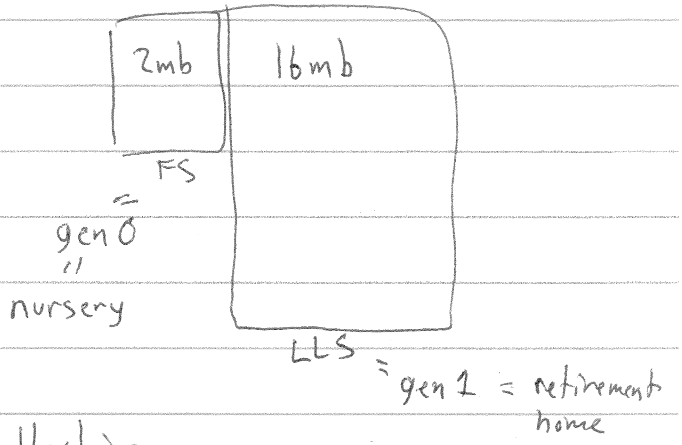


A forwarding pointer 

12-2) Long-lived things are copied a lot



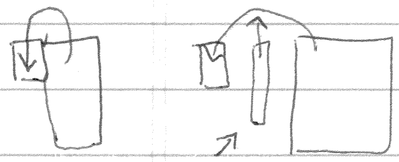




Generational Collection

"most objects die young"

do S+C often, do M+S sparingly



Inter-generational Pointer Table

```
ssss M ssss M
     ↓      ↓
   igpt 1  igpt 2
```

```
Program: o.f = x

now: if x is new
     && o is old
         igpt.add(o, x)
     o.f = x
```

write barrier

G.G.C: mem is as M+S & IGPT

time is as S+C + write barriers

Radioactive decay model