20-1/ Mark and Sweep

Time: malloc — $O(\lg n)$
free — X
gc — $O(\text{live}) + O(\text{mem})$
latency — use incremental collection

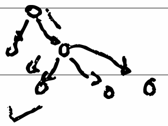
Space: overhead — mark bits = $O(\lg \text{mem})$

Stop and Copy

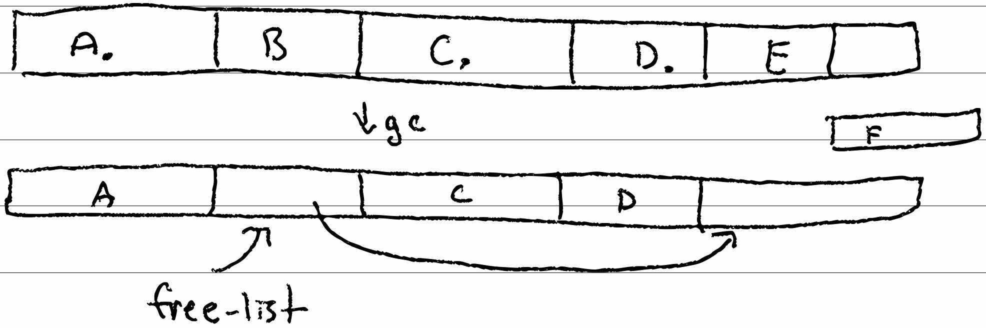
Time: malloc — $O(1)$
free — X
gc — $O(\text{live})$
latency — high, no incremental mode

Space: overhead — x 2

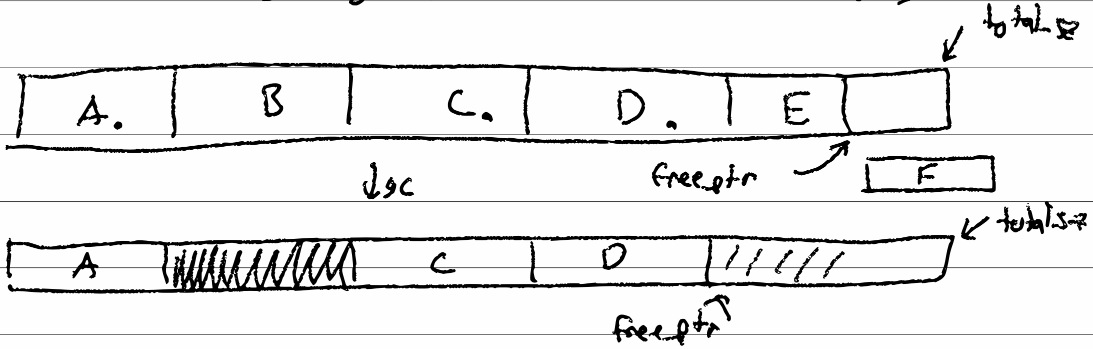
20-2) goal: O(1) malloc and O(live) gc

mls malloc searches a tree  O(lg n)

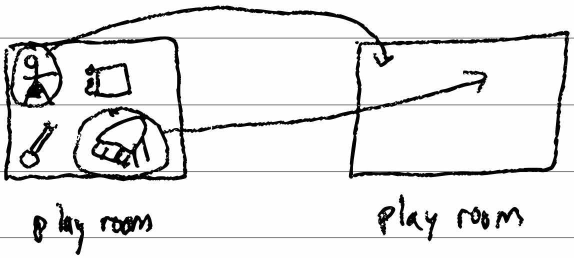
slc: O(1) ⟹ obvious where to malloc



```
        int free_ptr, total_sz;
20-3/   malloc (sz) {
           if (free_ptr + sz < total_sz) {
                 free_ptr += sz;
                 return (free_ptr - sz); }
           else { gc(); return malloc(sz); } }
```



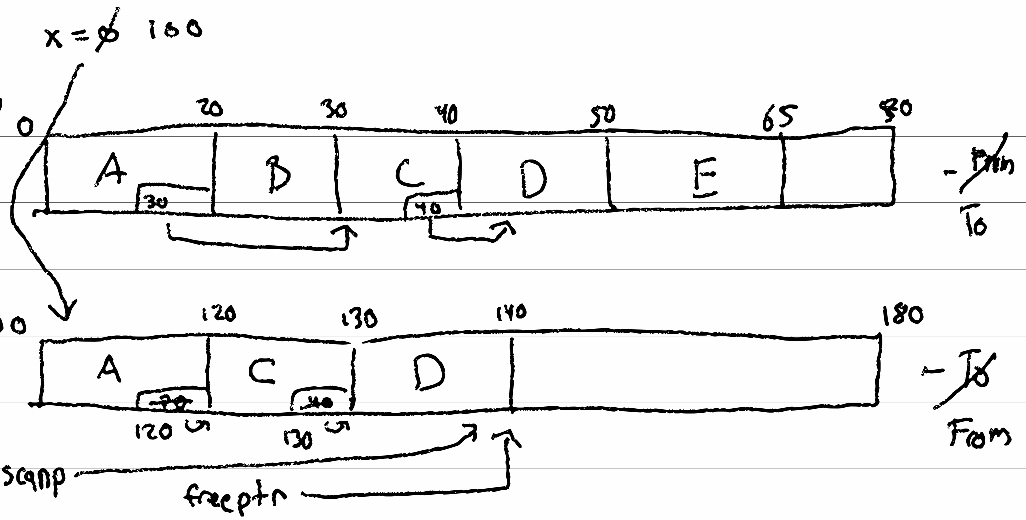
20-4/ why gc() is O(live)?



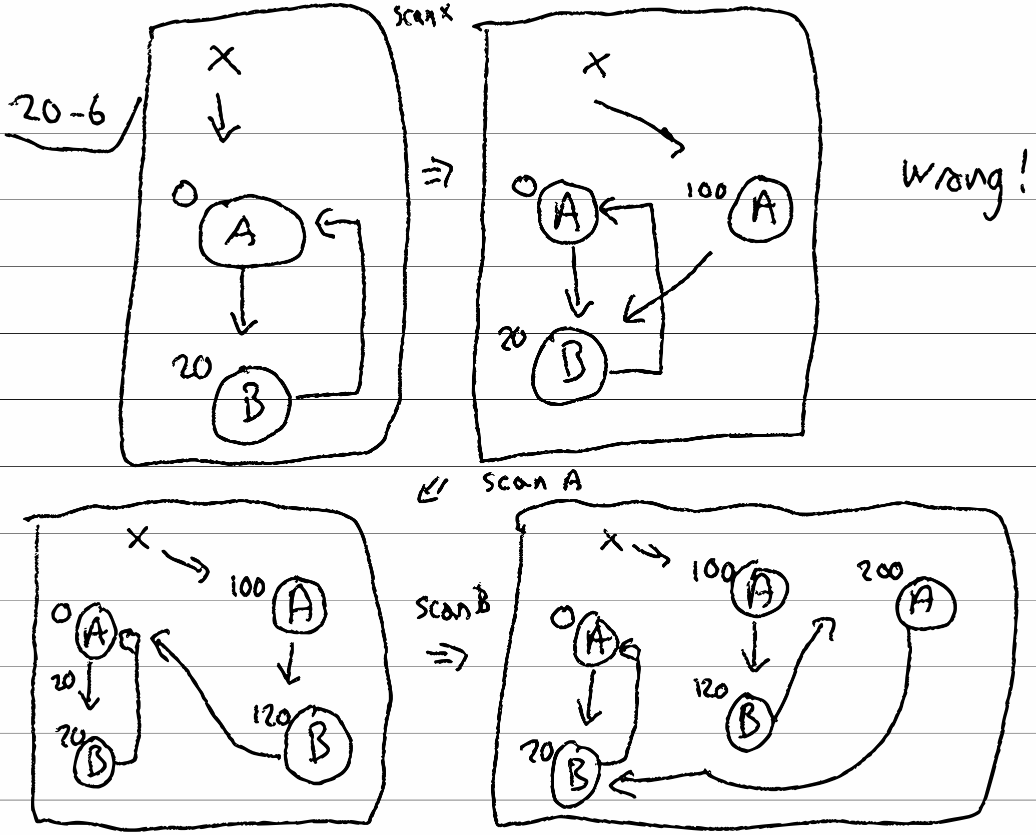
stop & copy

why spare x2

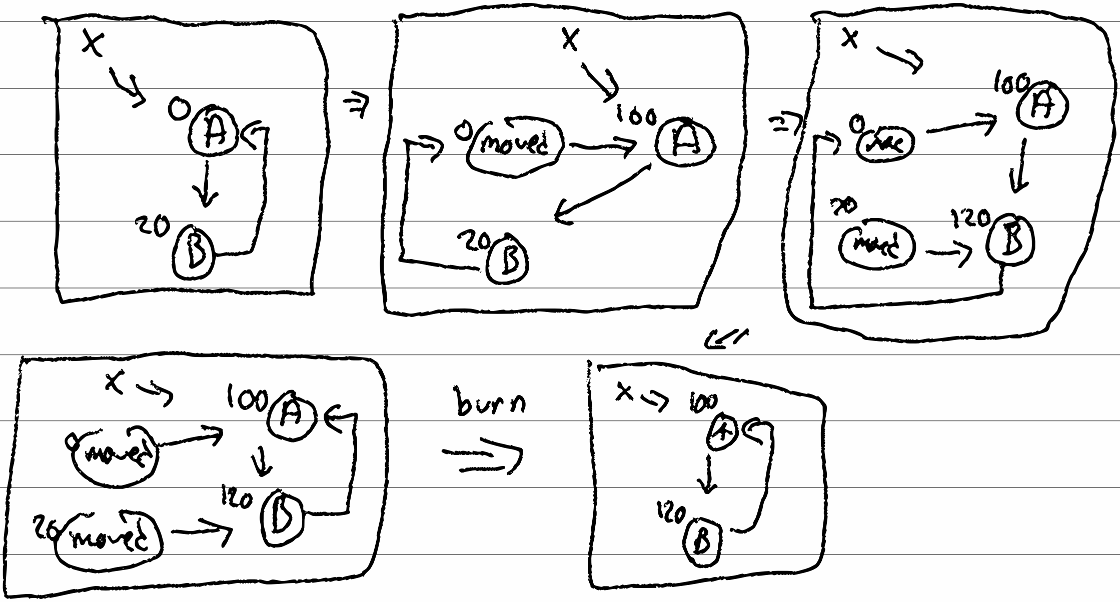
20-5



Cheney queue Implementation
of stop & copy


20-6
scan x
X
0
A
20
B
⇒
X
0
A
100
A
20
B
Wrong!
⇐ scan A
X
100
A
0
A
20
20
B
120
B
scan B
⇒
X
100
A
200
A
0
A
120
B
20
B

20-7/ After we copy, we update the old obj with a "forwarding" address



20-8/ How big is a forwarding pointer?

Tag and pointer

↓ ↓

always the same

word-sized (64-bits)

.... any object must be able to be changed into a forwarding pointer

.... therefore, all objects must be at least word-sized