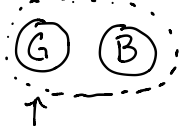
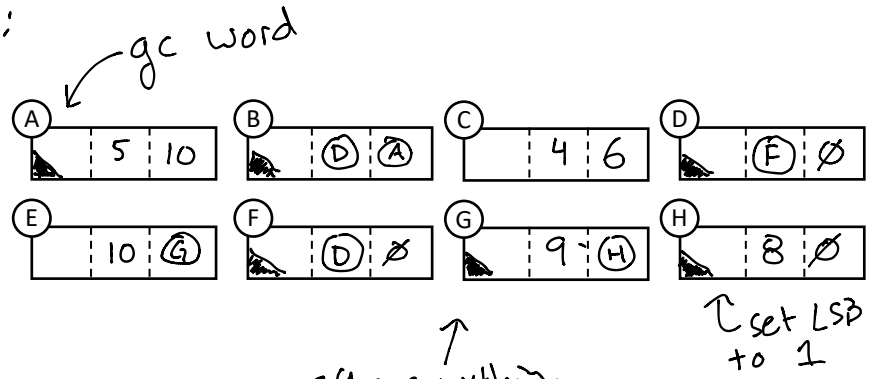
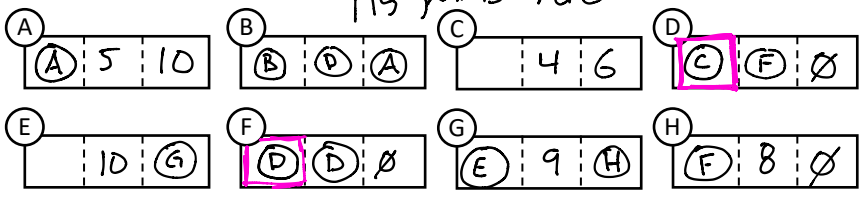


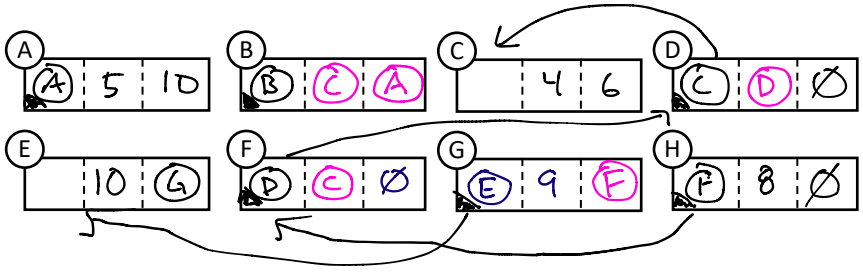


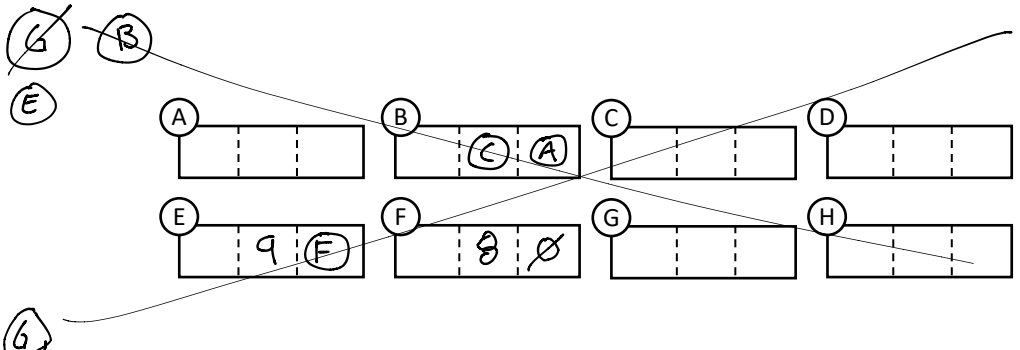
after everything I15 points here

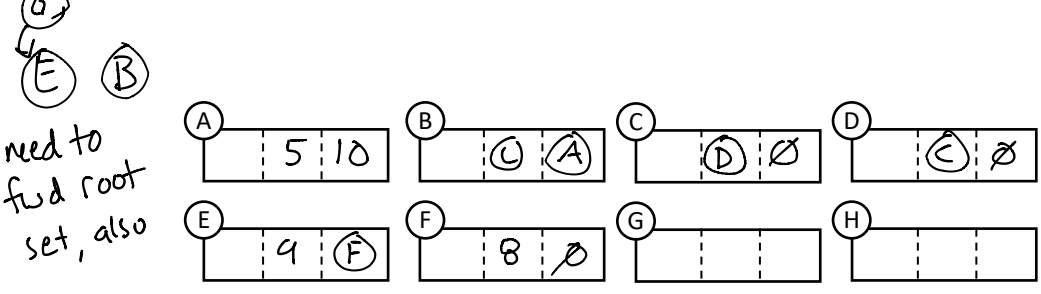


move_to = A B C D E F G

move_from = A B C D E F G H







need to fwd root set, also

MARK

```
for each ref r in root set
    mark_heap(r)

mark_heap(r):
    if *r == 1: return
    *r = 1
    for each ref r' in r:
        mark_heap(r')
    and save MAX_ADDR
    and save MIN_ADDR

Forward1():
    while move_from < MAX_ADDR
        if move_from→marked:
            *move_from == 1
            move_from→fwd = move_to
        move_to += 3 words
        move_from += 3 words
Forward2():
    traverse + update internal pointers

Compact():
    A: Traversal (mark)
    B: Iteration (fwd1)
```

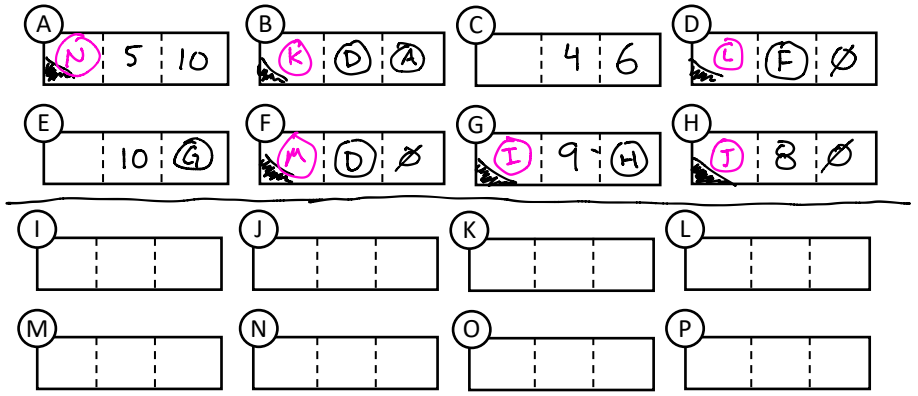
Mark/Compact

including "dead" data

O(HEAP SIZE)

~~O(LIVE DATA)~~

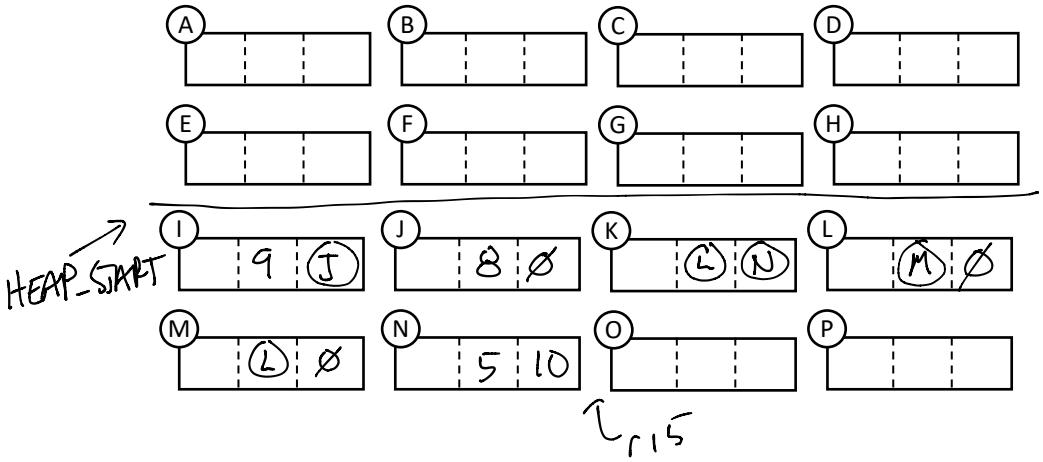
G B




```
mark():
    for each ref in root set
        mark_heap(ref)

mark_heap(ref)
  if *ref != 0: return
  *ref = next_addr()
  for each r' in refs:
      mark_heap(r')
```

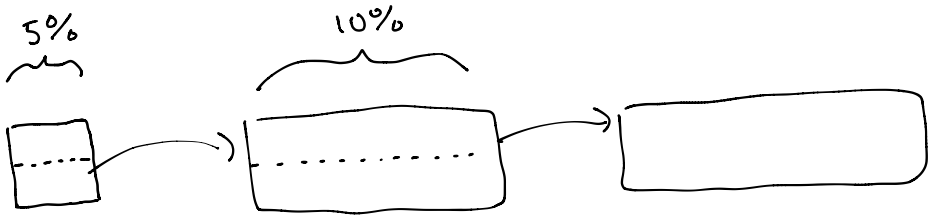
updating fwdg ptrs can be done w/traverse



SEMISPACE SWAPPING

STOP + COPY

O(LIVEDATA)



minor gc

major gc

stop + copy

count copies

tenured

mark/compact

free list

Eden

"Infant Mortality is high"

incremental

concurrent

GC