Eagle ⓪ Do nothing and hope

Gull ① When out of memory:
- identify heap objects not in use
- shift all in-use objects (live objects) left to shift all free memory right
- update heap_cursor

] garbage collection

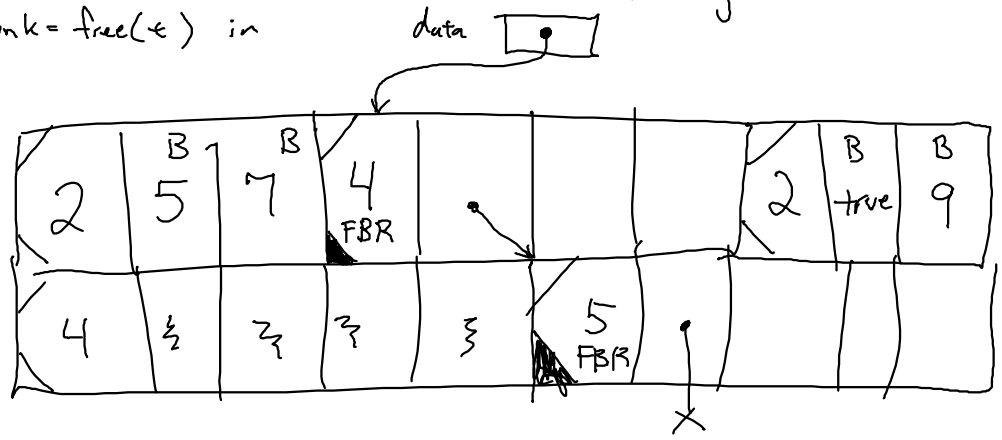
Requires:
- allocate GC words w/ each heap object
- how much memory is in use and size of heap
- distinguish pointers from other data

Guaiabero ② Assume each tuple/closure is like malloc

Add "free (<expr>)" to language

Manual Memory Management

```
let t = (5,6) in
let n = t[0] in
let junk = free(t) in
n
```



accounting structure:

track free blocks

initially: heap is all free

allocation carves pieces off

free creates new blocks and coalesces adjacent blocks