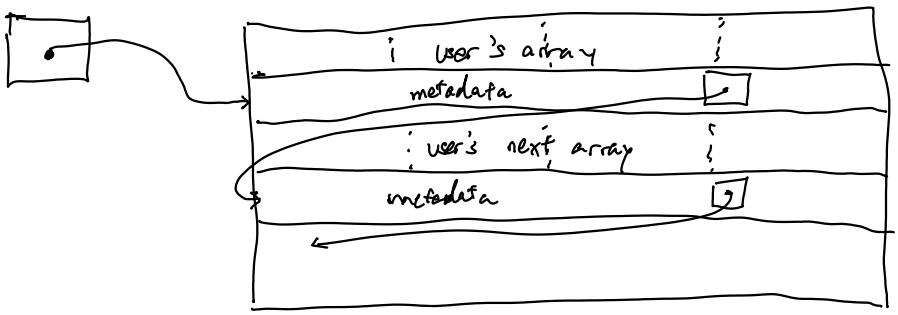
Memory Management

- What guarantees / responsibilites does programmer have w.r.t. memory?
- Who is responsible for allocation & deallocation?
- What form is metadata to track allocation?
- Semantics of language & allocation
- Allocation patterns that are problematic?

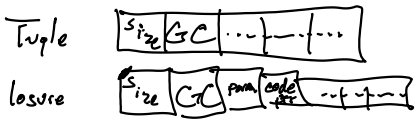


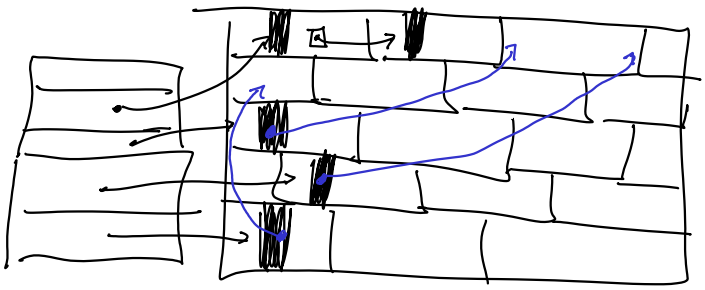
Automate Memory Management

* Eagle - not good
* (lots of options)
* Garbage collection (libraries part of language runtime)

Mark / Compact

- Start w/ heap region
- Cursor separate allocated / unallocated
  (also remember start & end of heap memory)
- Each allocation contains GC word





Only memory in use is accessible from stack (even indirectly)

0. Invariant: all GC words are 0
1. Mark everything reachable with GC = 1 (DFS)
2. Forward: update each GC word w/ new address (iterate)
3. Update: update every ptr to refer to its new address (iterate / DFS)
4. Compact: move everything to its forwarding address
5. Remark: set GC word to zero, update heap_cursor